

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)
UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) BARCELONATECH

Xarxa social per a hotels

Enginyeria Informàtica

Setembre 2016

Autor:

Antoni Colom Quetglas

Directora:

Claudia Patricia Ayala Martínez

Departament d'Enginyeria de Serveis i Sistemes d'Informació

Índex

1	Introducció i estat de l'art.....	5
1.1	Formulació del problema	5
1.2	Context	6
1.3	Estructura de la memòria.....	7
2	Abast del projecte	8
2.1	Obstacles	8
2.2	Metodologia de feina	9
2.3	Eines de seguiment	9
3	Planificació temporal	11
3.1	Descripció de les tasques i recursos.....	11
3.2	Diagrama de Gantt	12
3.3	Recursos humans	13
3.4	Valoració d'alternatives i pla d'acció	14
4	Gestió econòmica	15
4.1	Identificació i estimació dels costos.....	15
4.2	Control de gestió	16
5	Anàlisi inicial.....	18
5.1	Estudi de viabilitat	18
5.2	Anàlisi de riscos.....	22
5.3	Visió global de les funcionalitats del sistema.....	22
5.4	Anàlisi de tecnologies candidates.....	25
5.5	Conclusions de l'anàlisi inicial	26
6	Requisits.....	27
6.1	Propietats i hipòtesis sobre el domini.....	27
6.2	Restriccions	27
6.3	Diagrama de casos d'ús.....	28
6.4	Casos d'ús.....	35
6.5	Requisits no funcionals.....	66
7	Especificació.....	72
7.1	Esquema conceptual	72
7.2	Esquema del comportament.....	76

8	Disseny	95
8.1	Arquitectura	95
8.2	Diagrama de classes de disseny	100
8.3	Model de dades.....	101
8.4	Diagrames de seqüència	102
9	Implementació i Proves	106
9.1	Instal·lació d'un projecte Django	106
9.2	Creació i definició d'una aplicació Django	107
9.3	Proves.....	116
10	Conclusió.....	132
10.1	Desviacions en la planificació i pressupost.....	132
10.2	Diagrama de Gantt final	134
10.3	Limitacions i dificultats	135
10.4	Coneixements aplicats adquirits durant la carrera	136
10.5	Treball futur	136
11	Bibliografia	138

1 Introducció i estat de l'art

La finalitat d'aquest projecte es crear una xarxa social per a que l'experiència d'un viatge que s'allotja en un hotel sigui més completa, alhora que l'hotel rebi un feedback dels seus clients molt més realista. A més a més els comerços del voltant de l'hotel puguin publicitar-se en ell i els hostes en poden treure profit.

1.1 Formulació del problema

Actualment una de les mancances que trobem a un hotel és la falta de comunicació entre els hostes. Normalment només interactuen amb els que comparteixen habitació o viatge. Això ho sol·lucionarem afegint una interacció directe amb tots els clients de l'hotel a través de la nostra aplicació.

Actualment el client d'un hotel normalment arriba, utilitza els serveis de l'hotel i marxa. Els serveis més comuns són l'habitació i el restaurant. Addicionalment podem trobar-hi piscina, gimnàs, SPA, sauna, sales de reunions, pistes de joc, bar, discoteca... Amb aquest projecte volem fer dels clients, una comunitat; i afegir l'aplicació com a uns dels serveis bàsics de l'hotel.

D'altra banda, un hotel per comunicar-se amb el client utilitza cartells distribuïts per les diferents seccions de l'hotel o dins de l'habitació de cada client. Amb l'aplicació, el client podrà revisar totes les activitats que ofereix l'hotel durant la seva estada.

Normalment, quan realitzem un viatge ens informem de les possibilitats que ofereix l'entorn abans d'anar a la destinació o simplement investiguem la zona un cop ens trobem allà. Amb aquest projecte volem crear una eina perquè els comerços i activitats properes a l'hotel es publicitin oferint als clients descomptes o promocions diàries.

Avui en dia, quan un usuari publica una opinió d'un hotel a Internet (TripAdvisor, HolidayCheck, Booking...) s'ha de validar que aquell client ha estat allotjat allà. Amb l'aplicació no cal aquesta validació ja que l'usuari que ha estat a l'hotel podrà opinar de l'hotel quan vulgui sense validació, ja que si ha pogut utilitzar-la vol dir que l'hotel ha confirmat el seu registre.

1.1.1 Objectius

- L'objectiu principal d'aquest projecte és la creació d'una xarxa social per els clients allotjats a un hotel. El dividirem amb aquests subobjectius:
 - Anàlisi i especificació dels requeriments: Detectar els requeriments i necessitats del projecte i descriure'ls formalment.
 - Disseny i implementació de la base de dades.
 - Disseny i implementació de models i controladors.
 - Disseny i implementació del gestor de continguts.

- Disseny i implementació de l'interfície multi-dispositiu.
- Implementació i automatització de testeig unitari del codi.
- Realització de la documentació del funcionament de l'aplicació, la implementació del codi i del desplegament d'aquest.
- Un altre objectiu és l'adquisició de coneixements gràcies a la realització d'aquest projecte. Profunditzaré en tecnologies molt específiques, les quals haurem d'estudiar a fons, per poder dissenyar i implementar l'aplicació.

1.2 Context

La necessitat d'aquesta aplicació no és per a cap hotel en concret. Va ser una idea que va sorgir per a ser implementada a qualsevol hotel del món.

Un requisit bàsic és que l'hotel disposi de connexió a Internet. Si l'hotel disposa de WiFi de pagament la idea és que exclogui el domini de l'aplicació per poder-hi accedir gratuïtament.

Aquest projecte està pensat per ser implementat en hotels amb moltes persones hostatjades, com per exemple grans resorts o hotels turístics. No obstant, també pot funcionar en petits hotels o hostals ja que es poden aprofitar les funcions que no depenen de la interacció entre hostatjats.

Una altra opció serien les residències d'estudiants o fins i tot una comunitat de veïns.

El públic pel qual s'ha pensat l'aplicació és per a hotels turístics amb clients joves. L'aplicació és farà en anglès, preparada per assolir qualsevol idioma i serà aplicable a qualsevol localitat del món.

1.2.1 Parts interessades

Les parts interessades (o Stakeholders) són qualsevol individu, grup o organització que es vegi afectat per aquest projecte, obtenint així algun tipus de benefici, perjudici o impacte. Cada un d'ells pot influir de manera directe o indirecte al projecte.

Realitzador del projecte

En aquest cas soc jo mateix que realitzaré el projecte i la programació de l'aplicació que l'acompanya.

Tutor del projecte

Persona encarregada de fer el seguiment del projecte

Personal de l'hotel

Persones que treballen dins de l'hotel on s'utilitza l'aplicació i es veuran condicionades a utilitzar-la per tal de gestionar el contingut i els usuaris

Usuari

Persona que utilitzarà l'aplicació en benefici propi sense realitzar tasques administratives. Aquest ha de ser client de l'hotel.

Anunciant

Persona o empresa que vulgui publicitar-se dins de l'aplicació.

1.3 Estructura de la memòria

Estructuraré la memòria en varis capítols.

El primer es centrarà amb l'abast del projecte, obstacles, metodologia i eines de seguiment.

Tot seguit el de planificació, descripció de tasques, diagrama de Gantt i recursos humans, seguit d'una valoració d'alternatives i un pla d'acció.

A continuació dedicaré un capítol a parlar de la gestió econòmica. Faré una estimació de costos i un control de la gestió.

El següent capítol serà un anàlisi inicial que contindrà un estudi de viabilitat, anàlisi de riscos, una visió global de les funcionalitats del sistema i anàlisi de tecnologies candidates.

Continuaré amb un capítol per analitzar els requisits: hipòtesis sobre el domini, els casos d'us i els requisits no funcionals.

A continuació dedicaré un capítol a l'especificació amb l'esquema conceptual i el del comportament.

Tot seguit, continuaré amb el de disseny explicant l'arquitectura a seguir, el models de dades i els diagrames de seqüència.

Per acabar, faré un capítol per explicar la implementació i les proves realitzades.

2 Abast del projecte

Pel projecte es realitzaran les següents funcionalitats:

Aplicació web Responsive

Inicialment s'ha optat per realitzar una aplicació web responsive, és a dir, que s'adapti a tot tipus de pantalla per poder-la utilitzar a tots els dispositius: escriptori, tauletes i mòbils. Aquesta part inclou els models de dades, els controladors i les vistes.

Gestor de continguts

L'aplicació disposarà d'un gestor de continguts per tal de gestionar usuaris, permisos, variables de configuració, continguts de l'aplicació, etc.

Sistema multi-llenguatge

Es crearà tot el sistema pensat per ser traduït a qualsevol idioma, tot i que, la primera versió serà només en anglès.

S'han deixat de banda altres funcionalitats per a un desenvolupament posterior al projecte com podrien ser:

- Creació d'aplicacions natives per a Android i iOS.
- Creació d'una API REST Full per a comunicacions entre altres dispositius i base de dades.

Per falta de temps, recursos i experiència he decidit deixar fora les aplicacions natives de mòbil i centrar-me en una versió web multi dispositiu.

Donada aquesta decisió no em cal una API per comunicar el servidor en diferents dispositius. Per tant ha quedat fora de l'abast que s'havia planificat al principi.

2.1 Obstacles

Definirem els obstacles com a causes que provoquin un pobre funcionament de l'aplicació o l'impossibilitat d'utilitzar-la.

Un dels problemes que podríem tenir, seria la baixa acceptació o el rebuig dels hotels i això provocaria que els usuaris finals no la poguessin utilitzar.

Un altre obstacle seria el rebuig per parts dels usuaris que no és sentissin còmodes utilitzant l'aplicatiu.

Si els anunciants veuen que hi ha pocs usuaris també podrien decidir no utilitzar-la, fet que provocaria que els ingressos baixessin i que els usuaris no poguessin gaudir d'un dels beneficis de l'aplicació.

També es podrien donar un dels següents riscos definits amb la seva probabilitat i impacte.

Risc	Probabilitat	Impacte
Zona sense connexió a Internet	Mitja	Alt
Dispositiu amb baixa bateria	Mitja	Alt
Límit de consum de dades	Baixa	Alt
Servidor no respon	Baixa	Alt
Base de dades no respon	Baixa	Alt

2.2 Metodologia de feina

Per a realitzar el projecte faré servir el desenvolupament iteratiu.

Dividiré el projecte en iteracions. La primera iteració inclourà les parts de introducció, abast del projecte, planificació temporal, gestió econòmica i anàlisi inicial.

Les següents iteracions seran una part del producte final funcional. Les iteracions es realitzaran per ordre d'importància, segons funcionalitat, aquestes seran: Hotels, Usuaris, Opinions, Esdeveniments, Anuncis i Xat. Cada una d'elles tindrà les següents seccions: requisits, especificació, disseny, implementació i proves. Segons vagi avançant en les etapes podré retrocedir per tal de corregir els errors que em trobi.

Finalment realitzaré una iteració pel capítol de conclusió.

Els avantatges de seguir aquesta metodologia són els següents:

- L'usuari final pot tenir una versió parcial funcional del projecte al finalitzar cada iteració.
- Es redueixen els riscos del producte. A cada iteració es poden detectar errors que es corregeixen a la iteració següent. També dóna més control sobre la planificació ja que al final de cada iteració es pot replanificar la següent.
- Permet separar la complexitat del projecte

2.3 Eines de seguiment

Per tenir un control del que s'està fent a cada moment utilitzaré una taula de càlcul compartida al núvol (Google Sheet) amb el següent format.

ID	ParentID	Tasca	Estat	Estimació	Hores dedicades	Hores pendents	Comentaris

- **ID:** serà un número únic per a cada tasca per a poder-la identificar en qualsevol moment del projecte.
- **ParentID:** Identificador de la tasca pare si es tracte d'una subtasca.
- **Tasca:** serà el nom descriptiu de la tasca.
- **Estat:** Tindrà varis valors:
 - *Pendent:* Tasca que s'ha llistat però encara no ha estat planificada.

- *Planificada*: Tasca que s'ha estimat les hores de feina però encara no s'ha començat a treballar.
- *En curs*: Tasca que ja s'han invertit hores de feina i encara no s'ha acabat.
- *Finalitzada*: Tasca que ja s'ha realitzat.
- *Cancel·lada*: Tasca planificada que finalment s'ha decidit no realitzar.
- **Estimació**: Nombre d'hores que s'estima per realitzar aquesta tasca.
- **Hores dedicades**: Hores reals invertides en la realització de la tasca.
- **Hores pendents**: Hores que falten per acabar la tasca.
- **Comentaris**: Ampliació de la descripció de la tasca o comentaris sobre aquesta.

La idea és que cada dilluns es llistin totes les tasques que realitzaré aquella setmana i li doni una estimació en hores. Al final de la setmana haurien d'estar acabades i les hores dedicades haurien de ser iguals a les estimades. Si es produeix una diferencia molt elevada entre aquests dos nombres, s'hauria de replanificar el projecte (canvi de data d'entrega, eliminació de tasques poc rellevants...)

Per a versionar el codi font de l'aplicació utilitzarem GIT, allotjat a la plataforma Bitbucket. Això ens permetrà tenir una copia de seguretat del codi al núvol, tenir un seguiment de com evoluciona el codi i si algun dia hi ha més desenvolupadors, poder coordinar el codi entre nosaltres.

3 Planificació temporal

El projecte té una durada aproximada de 13 mesos, des de l'abril del 2015 fins al juny de 2016, ambdós inclosos.

3.1 Descripció de les tasques i recursos

El projecte s'ha dividit en 7 fases. Per a cada una d'elles s'especifica el rol que la realitzarà.

3.1.1 Fase I: Anàlisi previ

Rol: Cap del projecte

En aquesta fase es treballa la viabilitat del projecte avaluant els riscos que li podrien afectar una vegada posat en funcionament i es defineixen les parts interessades del sistema.

També estudiaré les tecnologies candidates que s'utilitzaran.

3.1.2 Fase II: Gestió de projectes

Rol: Cap del projecte

Aquesta fase està present durant tot el projecte. És la que engloba la part de gestió de tasques i seguiment del projecte.

3.1.3 Fase III: Anàlisi de requisits

Rol: Analista

En aquesta fase s'estableixen els requisits funcionals i no funcionals, les propietats i hipòtesis del domini i restriccions en les que es veurà involucrat el projecte. També es definiran els casos d'ús.

També donaré una visió global de totes les funcionalitats que podria arribar a tenir, seleccionant les que són imprescindibles pel projecte i descartant les que són menys importants i que no dispo del suficient temps per implementar-les.

3.1.4 Fase IV: Especificació

Rol: Analista

Realització de l'esquema conceptual i del model de comportament del sistema.

3.1.5 Fase V: Disseny, implementació i proves

Rol: Dissenyador

Pel que fa al disseny, es definirà la arquitectura del sistema, la distribució i funció de cada capa, els diagrames de seqüència de cada cas d'usuari, el diagrama de classes i la estructura de la base de dades.

3.1.6 Fase VI: Implementació i proves

Rol: Programador i tester

Fase en la qual es desenvolupa i testeja l'aplicació a partir de la feina realitzada pels altres rols en fases anteriors.

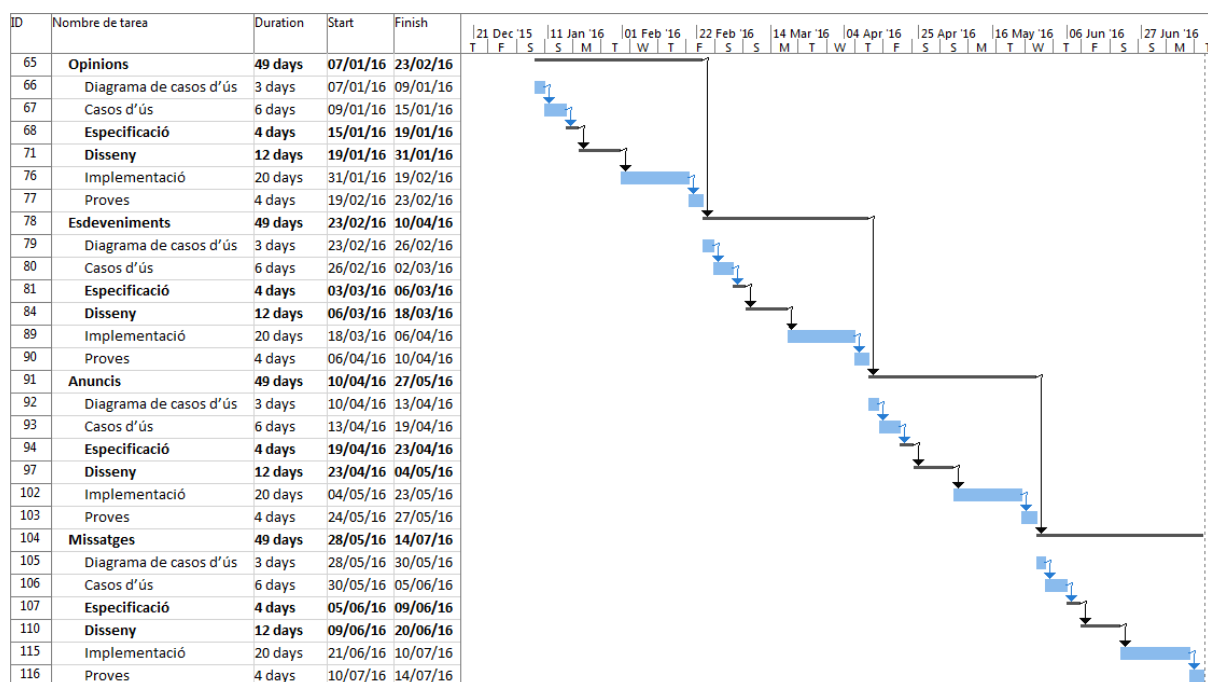
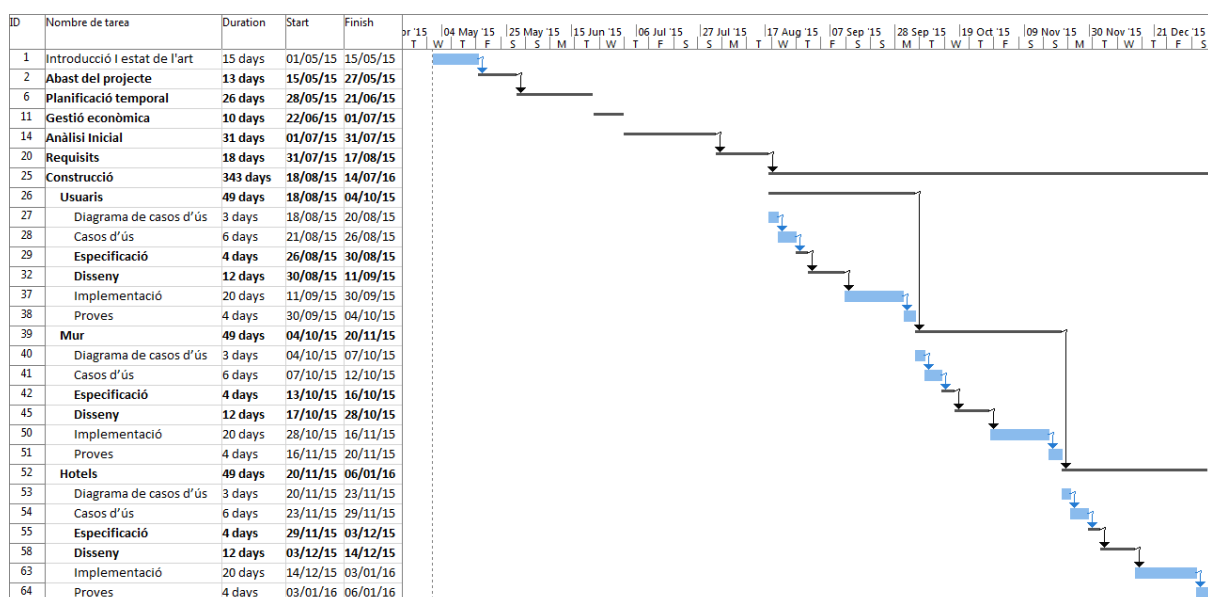
3.1.7 Fase VII: Final

Rol: Cap del projecte

En aquesta fase es revisarà tota la feina feta i es definiran els canvis que s'han realitzat sobre la proposta inicial.

3.2 Diagrama de Gantt

A continuació mostrem el diagrama de Gantt amb l'estimació inicial per la durada i realització del projecte.



En el [capítol de conclusió](#) podem trobar la versió final de com ha quedat el diagrama un cop acabat el projecte.

3.3 Recursos humans

Degut a que el projecte el desenvoluparà únicament una persona definirem els rols que adoptà durant tot el procés.

Cap de projecte

Encarregat de la organització, coordinació, supervisió i planificació del projecte.

Analista

Estudia les funcionalitats que tindrà el sistema a partir del model de casos d'us.

Dissenyador

A partir dels estudis realitzats per l'analista, dissenya les funcionalitats del sistema a desenvolupar.

Programador

Encarregat d'implementar una solució del disseny proposat. En aquest cas, també és l'encarregat de realitzar les proves unitàries i d'integració del seu codi.

3.4 Valoració d'alternatives i pla d'acció

Si alguna de les tasques es desvia disposarem d'un marge de temps per a suportar aquest desviament. En el cas que s'esgoti, realitzarem primer les funcionalitats principals de l'aplicació i deixarem pel final les menys importants o irrelevantes.

A continuació, llistem possibles riscos que ens podríem trobar un cop iniciada la fase de construcció.

Risc	Probabilitat	Impacte	Pla de contingència
Desviament en la planificació d'alguna de les fases	Alta	Alt	Realitzar revisions continues del procés del projecte. Dividir l'etapa de construcció en funcionalitats per ajustar l'estimació temporal i descartar funcionalitats menors. Augmentar el nombre de recursos, en aquest cas només disposem d'una persona, per tant es dedicaran més hores.
Canvis en la especificació d'alguna funcionalitat	Baixa	Alt	Aplicar molt bé el patró de Model – Vista - Controlador i separar les funcionalitats entre elles perquè el canvi d'una no afecti a les altres.
Desconeixement de la tecnologia	Baixa	Mig	Escollir tecnologies molt ben documentades i mantingudes per la comunitat. A poder ser utilitzades amb anterioritat.
Dependre de llibreries de tercers	Mitja	Mig	Modificar, dins del possible, la llibreria original perquè s'adapti a les nostres necessitats, en cas contrari, buscar una alternativa més assequible.
Alguna de les tecnologies utilitzades no és suficient per a realitzar alguna de les funcionalitats	Baixa	Alt	Estudiar alternatives que puguin substituir l'actual o que sigui complementaria. Un exemple seria si jQuery no és suficient, podríem utilitzar AngularJS.

En el [capítol de conclusió](#) expliquem quin riscs ens hem trobat i quin pla de contingència varem aplicar.

4 Gestió econòmica

4.1 Identificació i estimació dels costs

En aquest apartat desglossarem el cost del projecte en sous, software, hardware i costs generals.

4.1.1 Recursos humans

Ja que només disposem d'un recurs humà per realitzar el projecte dividirem els costos d'aquest segons els rols que exerceix.

Rol	Preu/hora	Hores	Preu total
Cap de projecte	30€	70	2.100€
Analista	20€	140	2.800€
Dissenyador	20€	80	1.600€
Programador	15€	180	2.700€
Total		470	9.200€

El càlcul de les hores has sortit del diagrama de Gantt. Estimo que tardaré 14 mesos en realitzar-lo, dedicant unes 8 hores setmanals aproximadament. Això fa una suma de 470 hores de feina.

4.1.2 Software

Aquí llistem tots els programes utilitzats durant el projecte. La majoria són de codi lliure i per tant tenen cost 0.

Producte	Cost projecte (14 mesos)
Microsoft Office	69€
Google Drive	0€
Microsoft Project (versió estudiant)	0€
Windows 7 (versió estudiant)	0€
ArgoUML	0€
SublimeText3 (versió de prova)	0€
Ubuntu 14.04	0€
Google Chrome	0€
Total	69€

4.1.3 Hardware

Llista de tots els elements hardware utilitzats per desenvolupar el projecte.

Producte	Cost	Període	Cost projecte (14 mesos)
Ordinador portàtil	700€	4 anys	175€
Smartphone	250€	2 anys	125€
Teclat + ratolí	40€	4 anys	10€
Monitor	200€	4 anys	50€

Total	1190€		350€
--------------	--------------	--	-------------

4.1.4 Costs generals

Llistem altres costs del projecte que no podem classificar als altres apartats.

Producte	Cost	Període	Preu
Servidor	10€/mes	1 mes	10€
Domini (.com + wilcard para subdominis)	12€/any	1 any	12€
Connexió a Internet ADSL	35€/mes	12 mesos	420€
Connexió a Internet mòbil	10€/mes	12 mesos	120€
Total			562€

4.1.5 Costs imprevists

Es poden produir varis costs imprevists com per exemple que el portàtil, algun perifèric o l'smartphone pateixin alguna avaria. El cost de la seva reparació dependria de la gravetat de l'incident.

Un altre imprevist seria un excés de transferència de dades en un dels servidors, el qual el cost s'hauria d'abonar apart i dependria del nombre de dades transferides.

4.1.6 Cost total del projecte

A continuació fem una taula resum amb el cost total del projecte.

Concepte	Preu
Recursos humans	9.200€
Software	69€
Hardware	452€
Costos generals	562€
Total	10.283€

4.2 Control de gestió

Per poder gestionar les hores que dedica cada rol en realitzar una tasca, s'anirà actualitzant una fulla de càlcul a núvol (Google Sheet) amb el següent format

Data	Cap de projecte	Analista	Dissenyador	Programador
23/07/2015	8h	2h	1h	5h

Al [final del projecte](#) es farà una taula resum de les hores que ha dedicat cada recurs i les hores estimades inicialment.

Recurs	Hores estimades	Hores dedicades
Cap de projecte	70h	
Analista	140h	

Dissenyador	80h	
Programador	180h	

També s'ha d'incloure un 15% de risc sobre l'estimació del projecte. Aquest risc s'utilitzarà per a imprevists o canvis de disseny no planejats inicialment.

5 Anàlisi inicial

Els factors fonamentals per a la realització del projecte han estat:

- Aportació de la idea per part d'un company.
- Experiència laboral en el sector hotelier.
- Facilitat d'accés a un hotel per poder provar-ho en un escenari real.
- Coneixement avançat de les tecnologies implicades.
- Motivació personal per emprendre.

5.1 Estudi de viabilitat

5.1.1 Aplicacions similars o alternatives del mercat

A continuació analitzaré aplicacions en les quals identifiquem alguna funció similar a les que proposem. D'aquesta manera puc aprendre del seu èxit, així com intentar millorar les mancances que hi trobem

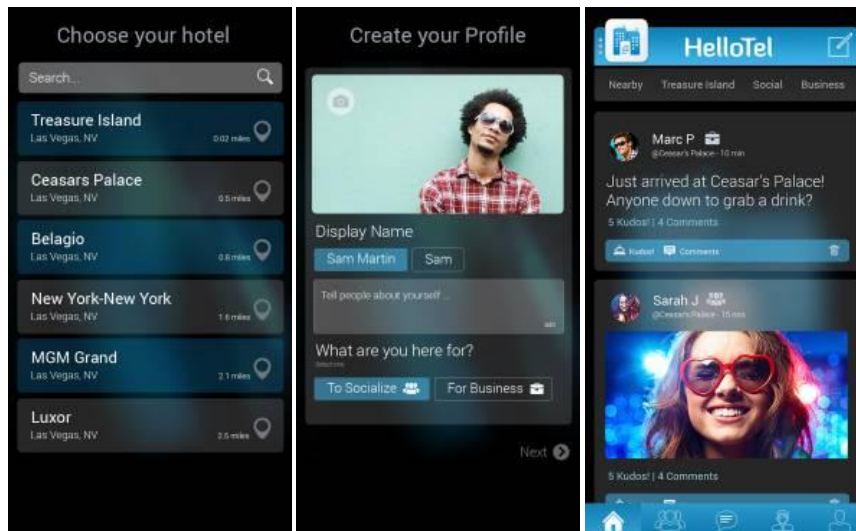
5.1.1.1 HelloTel

HelloTel (<http://www.hellotelapp.com/>) potser es l'aplicació amb més semblances amb aquest projecte.

Un cop registrats, ens demana a quin hotel ens allotgen per poder comunicar-nos amb la gent de l'hotel o de la zona. No he pogut provar cap funcionalitat ja que no hi ha cap usuari actiu.

El que no aporta aquesta aplicació és informació sobre el propi hotel on estàs allotjat (o et penses allotjar en un futur pròxim). La meua proposta inclou interacció per part de l'hotel ja que disposarà d'un panell d'administració on podrà publicar notícies, menús, excursions, crear esdeveniments, parlar amb altres usuaris... També es podrà veure tot la informació de l'hotel i les opinions dels clients.

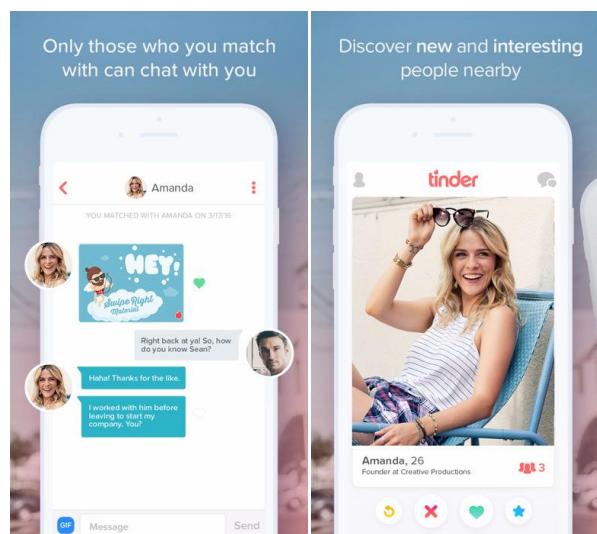
Per altra banda tampoc podem saber res de la zona on es situa l'hotel com ara restaurants, establiments o zones d'oci. Gràcies al sistema d'anuncis i esdeveniments podrem aportar aquesta informació a l'usuari.



5.1.1.2 Tinder

Tinder (<https://www.gotinder.com/>) a dia d'avui és l'aplicació de cites més utilitzada a tot el món. La idea és contactar amb persones que es troben dins d'un radi determinat i poder seleccionar o descartar els perfils amb un fàcil moviment de dit.

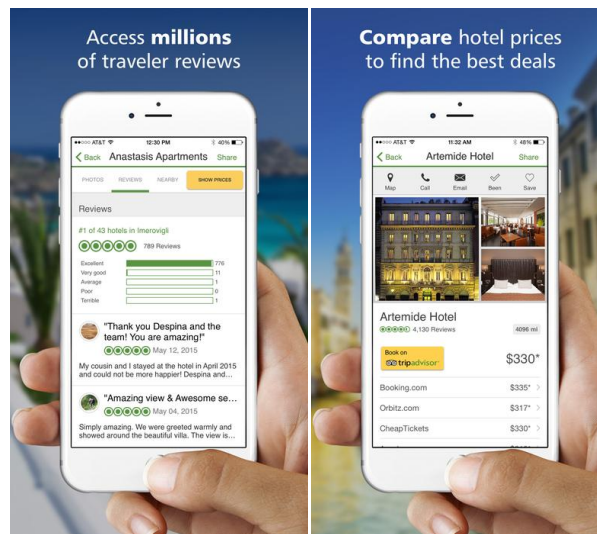
Degut a que utilitza la ubicació del mòbil per a trobar la gent del voltant amb com a mínim 2km de distancia, fa bastant impossible contactar amb algú del mateix hotel en un temps raonable.



5.1.1.3 TripAdvisor

TripAdvisor (<https://www.tripadvisor.es/>) és l'aplicació referent en quant a opinions d'hotels. Degut a la quantitat d'usuaris es pot obtenir una informació molt fiable sobre l'hotel. També disposa de tota la informació de l'hotel inclús permet fer reserves.

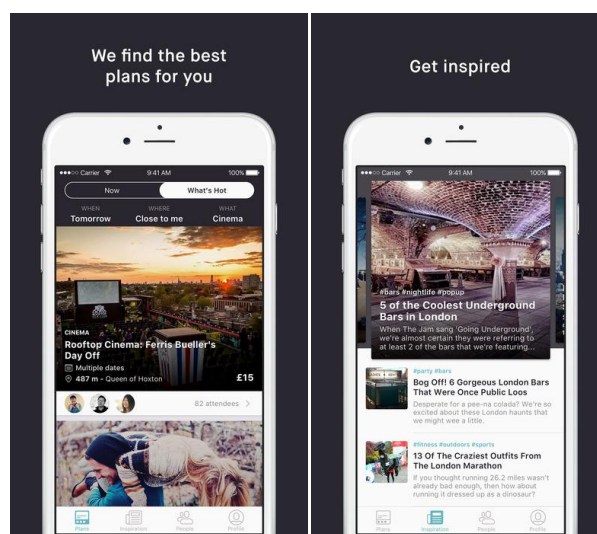
L'inconvenient que hi trobem és que les opinions dels usuaris o bé no estan validades, o bé per validar-les, l'usuari ha de demostrar que ha estat a l'hotel enviant una còpia de la seva reserva. Aquest projecte facilitarà aquest procés ja que només podran fer opinions els usuaris que realment han estat a l'hotel.



5.1.1.4 Fever

Fever (<https://www.feverup.com/>) ens ofereix esdeveniments que succeeixen en una ciutat o zona determinada.

La manca de l'aplicació són els esdeveniments més reduïts o simples com per exemple fer una excursió o compartir un cotxe per anar a la platja. En aquest projecte intentarem que l'usuari proposi activitats per poder relacionar-se amb altre gent de l'hotel.



5.1.2 Anàlisi de les parts interessades (stakeholders)

Client hostatjat a un hotel

Cada hotel tindrà dos tipus de clients, aquells que vulguin utilitzar l'aplicació i aquells que no. Nosaltres només ens centrarem en els que la utilitzen. Aquets usuaris s'hauran de registrar a l'aplicació i un cop validats podran accedir a tots els serveis que l'ofereix.

Administrador de l'aplicació

És la persona encarregada d'administrar els continguts de l'aplicació del seu hotel, gestionar els usuaris, opinions, etc. Idealment hauria de ser alguna persona que treballés a l'hotel, com per exemple a recepció. Si no és possible, hauria de ser alguna persona amb accés a les reserves de l'hotel per poder validar si l'usuari que es registre està a hostatjat a l'hotel o no.

Propietari/director/gerent d'un hotel

Persona que pren la decisió per a incorporar l'aplicació al seu hotel.

Comerciant de la zona

Comerç de qualsevol tipus interessat en anunciar-se o oferir descomptes a l'aplicació. Rep el benefici d'apropar-se a un nombre més elevat de clients.

Futur client d'un hotel

És un usuari interessat en allotjar-se a l'hotel que disposa d'aquesta aplicació i el qual consulta les opinions públiques que han realitzat els usuaris que ja han utilitzat l'aplicació. Gràcies a aquestes opinions, l'usuari pot conèixer el tipus de serveis que ofereix l'hotel.

Super usuari de l'aplicació

Usuari encarregat de gestionar l'aplicació. Crear nous hotels, activar nous serveis, canviar les variables de configuració, entre d'altres.

Per tal de tenir una idea de l'acceptació d'aquest projecte, he consultat a varies persones si la idea del projecte era atractiva per ells o no, preguntant a una mostra de joves d'entre 20 i 30 anys. La opinió generalitzada, ha estat que estarien molt interessats en utilitzar-la, sobretot quan fan viatges d'oci.

També ho he consultat amb un director d'hotel el qual és va mostrar molt interessat en tenir un sistema en el que pugui rebre feedback constant i directe dels seus clients i que a més a més permeti que el seu personal autoritzat es pugui comunicar amb els clients directament a través d'una aplicació.

5.1.3 Limitacions legals implicades

- L'aplicació constarà d'una política de privacitat i un avís legal que l'usuari haurà d'acceptar un cop es doni d'alta. Ambdós arxius seran sempre accessibles per l'usuari i se l'informarà cada cop que es produeixi una actualització per tal de tornar-les a acceptar o denegar.

- L'usuari també haurà d'acceptar una clàusula que asseguri que tot el contingut multimèdia que publiqui sigui original o respecti la llicència del contingut original. De no fer-ho, ell serà el responsable.
- Tot el contingut que publiqui l'usuari només serà vist per la resta d'usuaris registrats a l'aplicació, a excepció de les opinions de l'hotel que seran públiques.
- El personal de l'hotel no tindrà accés a les converses que realitzin en privat els usuaris.

5.2 Anàlisis de riscos

A l'apartat d'obstacles ja hem llistat els riscos del projecte, aquí veurem com mitigar-los i el pla de contingència.

5.2.1 Estratègia de mitigació

Límit de consum de dades

Per tal d'evitar que l'usuari consumeixi un excessiu nombre de dades tant la API com els recursos que sol·liciti l'aplicació seran mínims i sempre estaran comprimits al màxim possible.

Servidor no respon

En fet que el servidor no respongués podria ser degut a diverses causes. Per evitar-les, hostejarem el servidor a les infraestructures d'Amazon Web Services per tal de tenir el mínim de *downtime* possible.

Base de dades no respon

Per evitar que la base de dades no respongui també la hostejarem a Amazon Web Services . Intentarem minimitzar el nombre de lectures amb la utilització de caches i les que s'arribin a fer seran el més òptimes possible, limitant el nombre de registres que retornen i utilitzant índexs.

5.2.2 Pla de contingència

Servidor no respon

Si el servidor no respon l'aplicació mostrarà un avis. Automàticament s'avisarà a l'encarregat del manteniment per solucionar-ho en el mínim temps possible.

Base de dades no respon

Si la base de dades no respon l'aplicació mostrarà un avis. Automàticament s'avisarà a l'encarregat del manteniment per solucionar-ho en el mínim temps possible.

Falta de connexió a Internet

Si es detecta una pèrdua de la connexió s'avisarà a l'usuari amb un missatge.

5.3 Visió global de les funcionalitats del sistema

Dividirem les funcionalitats amb dos parts diferenciades, la part d'usuari i la part d'administració.

La part d'usuari està pensada per ser utilitzada des d'un ordinador, tauleta o dispositiu mòbil, en canvi la part d'administració està pensada per ser utilitzada només des de un ordinador o una tauleta.

5.3.1 Funcionalitats de la part d'usuari

La part de l'usuari constarà de dues parts, la part pública i la privada.

5.3.1.1 Part pública

Secció de l'aplicació que hi poden accedir tot tipus d'usuaris. Aquets podran fer:

Llistat d'hotels

Es mostrarà un llistat amb tots els hotels que tenen l'aplicació disponible.

Detall de l'hotel

Es veurà una descripció de l'hotel, ubicació, fotos, serveis i llistat d'opinions d'usuaris de l'aplicació.

Llistat d'opinions

Llista d'opinions que han realitzat els usuaris sobre l'hotel

Formulari de registre

Formulari on l'usuari introdueix les seves dades per ser validades posteriorment i poder utilitzar l'aplicació de l'hotel seleccionat.

Formulari de login

Formulari on l'usuari introdueix les seves dades d'accés a l'aplicació.

Formulari de contacte

Formulari per a que un usuari pugui demanar informació als gestors de l'aplicació. Normalment l'utilitzaran els hotels per donar-se d'alta.

5.3.1.2 Part privada

Secció de la web que només és visible als usuaris registrats, validats i logats. Aquests, a més a més de les funcionalitats de la part pública, podran veure:

Llistat de allotjats

Llistat amb els allotjats a l'hotel, que seran els usuaris que també s'han donat d'alta i han estat validats.

Fitxa de l'usuari

Detall d'un usuari amb les seves dades personals, fotos, publicacions.

Llistat de converses

Llistat de les converses que ha iniciat un usuari amb un altre.

Detall de la conversa o xat

Llista de missatges que s'han intercanviat dos usuaris.

Llistat de publicacions

Llistat de les publicacions que fa un usuari o el personal de l'hotel.

Detall d'una publicació

Detall de la publicació amb el nom de l'autor, el comentari, la data, arxius multimèdia (si en té) i llistat de comentaris sobre la publicació.

Llistat d'esdeveniments

Llistat d'esdeveniments publicats per un altre usuari, membre de l'hotel o anunciant.

Detall de l'esdeveniment

Detall d'un esdeveniment amb tota la informació. Data, descripció, assistents, multimèdia...

Anuncis

Bloc d'informació sobre un anunciant extern.

Mur

Llistat de publicacions, esdeveniments i anuncis barrejats.

5.3.2 Funcionalitats de la part d'administració

Aquesta part només podran accedir usuaris administradors. Depenent del permisos que tinguin podran administrar més o menys seccions.

Llistat i fitxa d'hotel

Llista d'hotels donats d'alta a la web amb els seus filtres. Cada un amb la seva fitxa per crear-lo, editar-lo o eliminar-lo.

Llistat i fitxa d'usuari

Llista d'usuaris donats d'alta a la web amb els seus filtres. Cada un amb la seva fitxa per crear-lo, editar-lo o eliminar-lo.

Llistat i fitxa d'opinions

Llista d'opinions enregistrades a la web amb els seus filtres. Cada una amb la seva fitxa per crear-la, editar-la o eliminar-la.

Llistat i fitxa d'esdeveniments

Llista d'esdeveniments enregistrades a la web amb els seus filtres. Cada una amb la seva fitxa per crear-lo, editar-lo o eliminar-lo.

Llistat i fitxa de contacte

Llista de contactes enregistrades a la web amb els seus filtres. Cada una amb la seva fitxa per crear-lo, editar-lo, eliminar-lo o respondre.

Llistat i fitxa d'anunci

Llista d'anunci donats d'alta a la web amb els seus filtres. Cada un amb la seva fitxa per crear-lo, editar-lo o eliminar-lo.

Llistat i fitxa de publicacions

Llista de publicacions fetes per un usuari a la web amb els seus filtres. Cada un amb la seva fitxa per crear-la, editar-la o eliminar-la.

5.4 Anàlisis de tecnologies candidates

Diferenciarem 3 nivells de tecnologia, *frontend*, *backend* i servidor.

S'ha decidit fer el *frontend* completament separat del *backend* per poder anar evolucionant o canviant el *frontend* sempre que sigui necessari i alhora poder tenir el *backend* separat amb possibilitats d'escalar.

La part de *frontend* i *backend* es farà amb *Django Framework*. *Django* és un conjunt de llibreries fetes amb *Python* que ens permeten construir un sistema web seguint el patró de Model - Vista - Controlador.

5.4.1 Backend

La part que considerarem *backend* seran les capes de Model i Controlador. *Django* consta d'un ORM (*Object-Relational Mapping*) que ens permet definir les taules de la base de dades a partir de classes escrites en *Python*. Això ens facilita la capa Model ja que podem realitzar modificacions a la base de dades sense preocupar-nos de quin motor hi ha al darrera.

La capa Controlador estarà format per les *Views* i *Forms* de *Django*. La primera s'encarrega de rebre, processar i retornar peticions HTTP, mentre que la segona s'encarrega de la lògica de formularis i validacions de camps.

5.4.2 Frontend

La part de *frontend* serà el model Vista. En aquest cas seran fitxers *HTML*, *CSS* i *JavaScript*. Per a l'*HTML* s'ha decidit fer servir el sistema de *Django Templates* ja que ens permet realitzar la lògica i el propi *framework* fa un procés de renderitzat abans de retornar l'*HTML*.

S'aplicaran estils per a obtenir una web responsive utilitzant Bootstrap.

Finalment, utilitzarem *jQuery* per a fer tota la lògica *frontend* amb *JavaScript*. Aquest s'encarregarà de fer algunes de les crides al *backend* utilitzant *AJAX*.

En una segona fase es podria fer una senzilla aplicació en *Android* que utilitzes el navegador natiu per mostrar l'aplicació.

5.4.3 Servidor

A la banda del servidor tindrem:

- Servidor web amb *NginX*.
- Base de dades amb *MySQL*.
- Cache amb *Redis*.
- Servidor de emails amb *Postfix*.
- Servei de cues per a tasques asíncrones amb *RabbitMQ*.

A més a més, es farà servir GIT com a control de versions del codi

5.5 Conclusions de l'anàlisi inicial

Per veure la viabilitat d'aquest projecte faré una taula comparativa entre les característiques de les aplicacions analitzades prèviament:

Aplicació	HelloTel	Tinder	TripAdvisor	Fever
Disseny i UX	Limitat	Excel·lent	Bo	Bo
Parlar amb gent d'un hotel	Si	No	No	No
Opinions d'hotels	No	No	Si	No
Esdeveniments	No	No	No	Si
Xat	Si	Si	No	No
Informació sobre comerços	No	No	Si	Si

De moment no existeix cap aplicació similar i per tant pot ser un atractiu tant pels usuaris com pels hotelers. L'objectiu es crear una aplicació amb el millor de cada una.

Degut a l'alt domini de les tecnologies implicades i a la utilització d'una arquitectura desacoblada, permet que el canvi d'una de les parts no afecti a la resta.

Finalment, tenir coneixement del sector hotelier i poder provar-ho en un entorn real, fa que el projecte sigui molt viable.

6 Requisits

6.1 Propietats i hipòtesis sobre el domini

6.1.1 Hipòtesis del domini

Connexió a Internet

Els usuaris han d'estar connectats a una xarxa d'Internet, ja sigui mòbil o Wifi.

Hotel

Com a mínim un hotel ha d'accedir a donar-nos accés per poder utilitzar l'aplicació ja que un membre del personal ha de confirmar l'alta de l'usuari.

6.1.2 Expectatives

Altes d'usuari legítimes

S'espera que els usuaris que es donin d'alta en un dels hotels estiguin allotjats en aquell hotel i en aquell moment o ho vagin a fer en els propers dies.

Utilització del sistema

S'espera que els usuaris i els administradors facin un ús correcte de l'aplicació. De totes maneres s'aplicaran mesures de seguretat perquè no es produeixin accions malicioses que afectin a les dades d'altres usuaris.

6.2 Restriccions

6.2.1 Restricció de solució

Descripció

La versió mòbil inicialment només estarà disponible per a versió web, en un idioma (anglès) i el disseny serà limitat prioritzant les funcionalitats.

Justificació

Per assolir totes les funcionalitats s'ha decidit implementar inicialment una web responsive. Es podran afegir més idiomes i el disseny serà modificat en una futura iteració fora d'aquest projecte.

6.2.2 Restricció temporal

Descripció

El projecte serà entregat durant el tercer trimestre del 2016.

Justificació

S'ha d'establir una data límit per a l'entrega del projecte.

6.2.3 Restricció econòmica

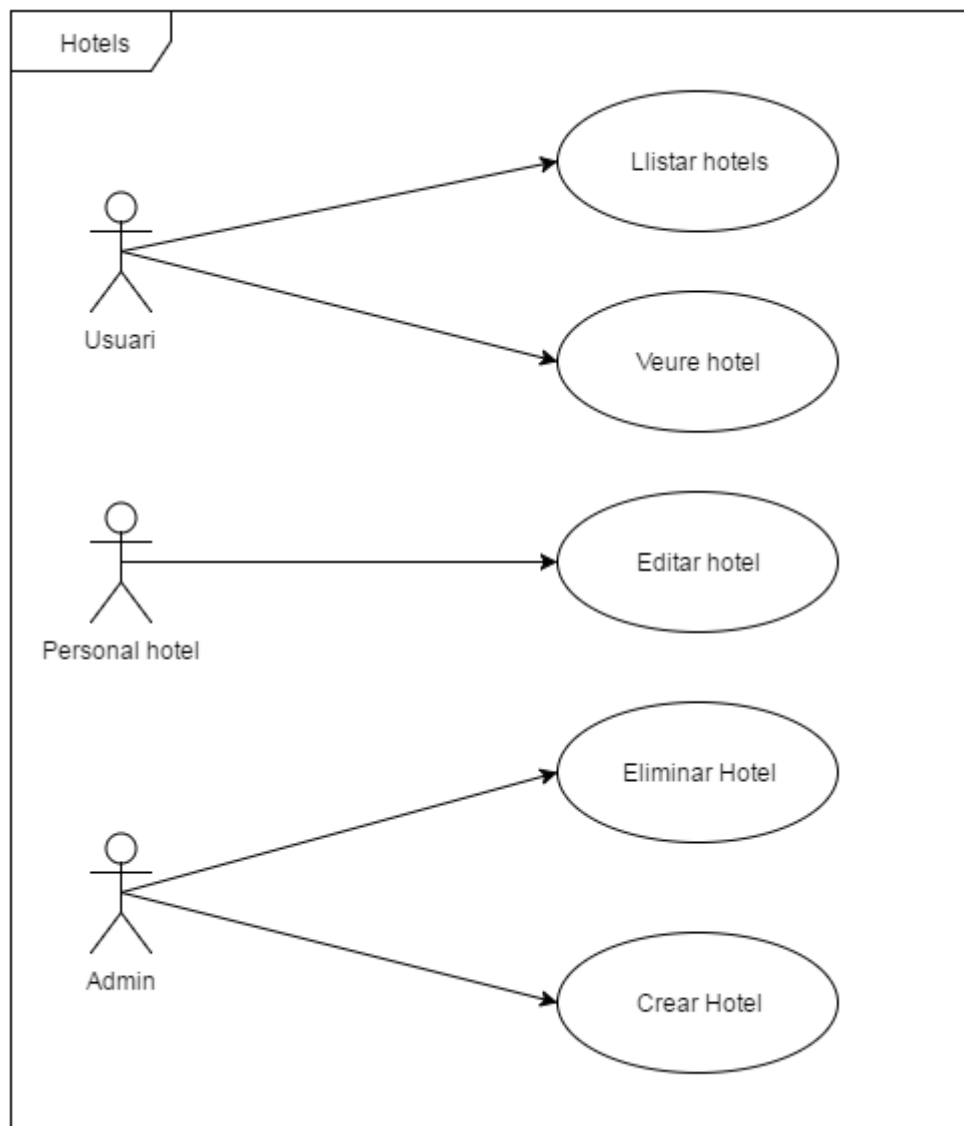
Descripció

El projecte es desenvoluparà sense cap suport econòmic.

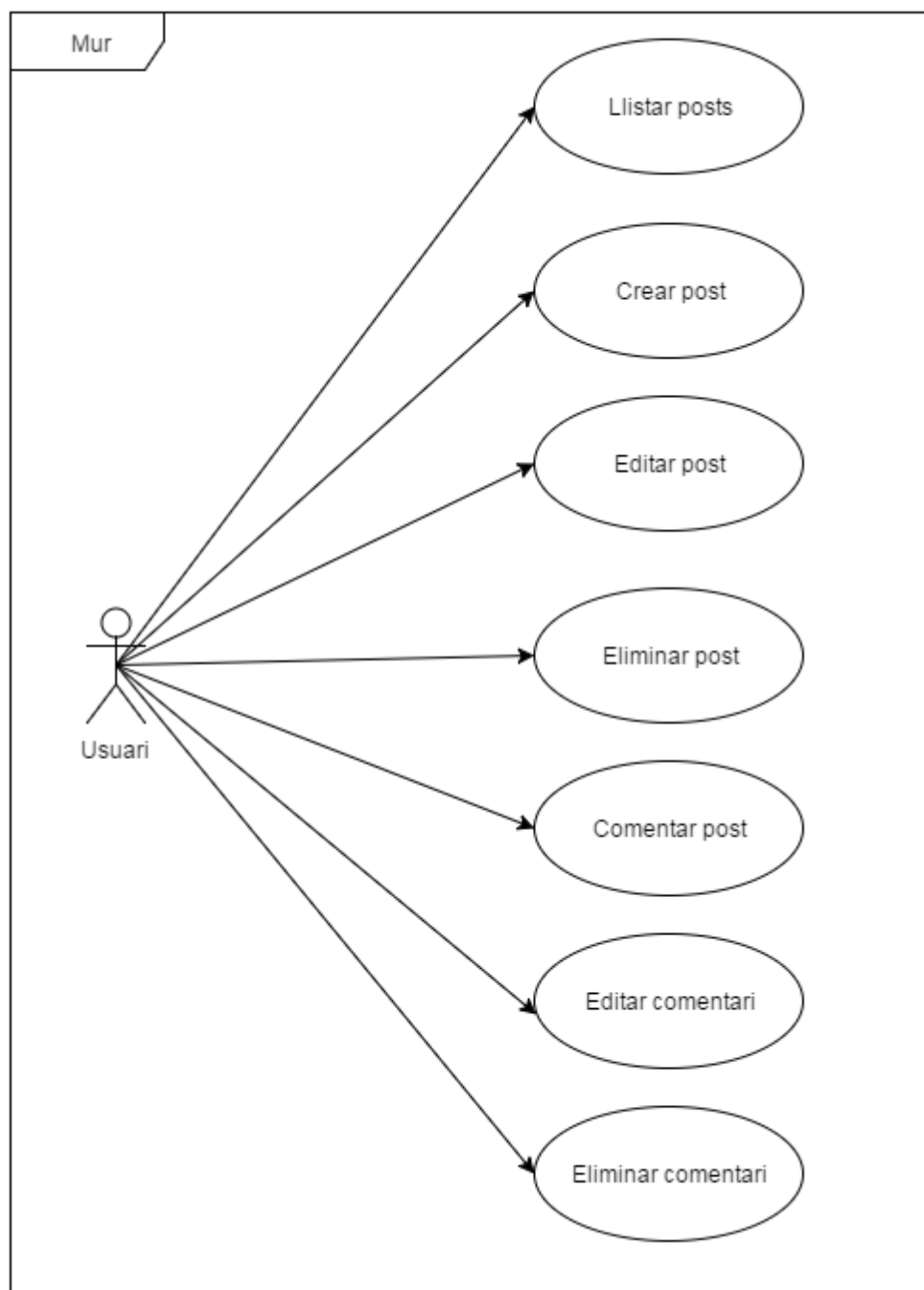
Justificació

El projecte només té finalitats acadèmiques.

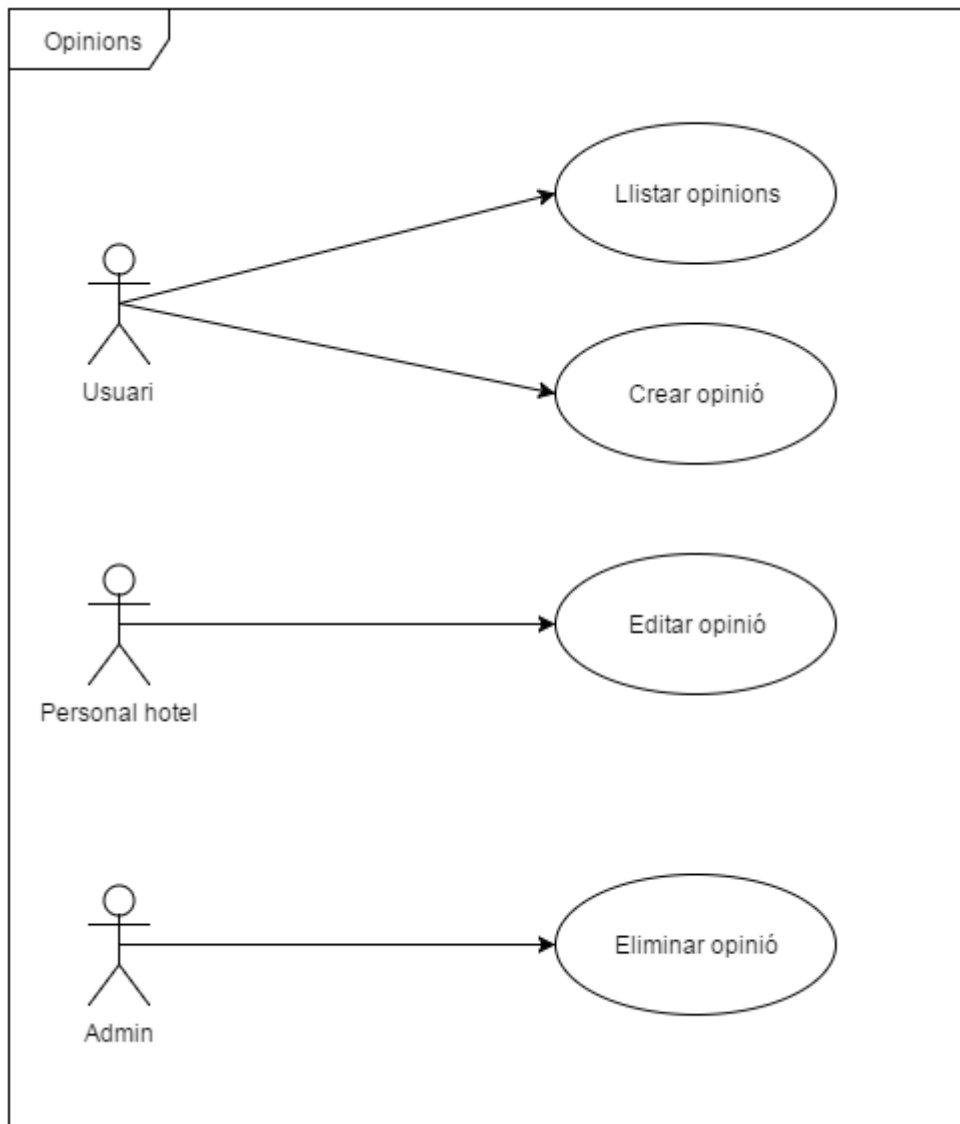
6.3.2 Hotels



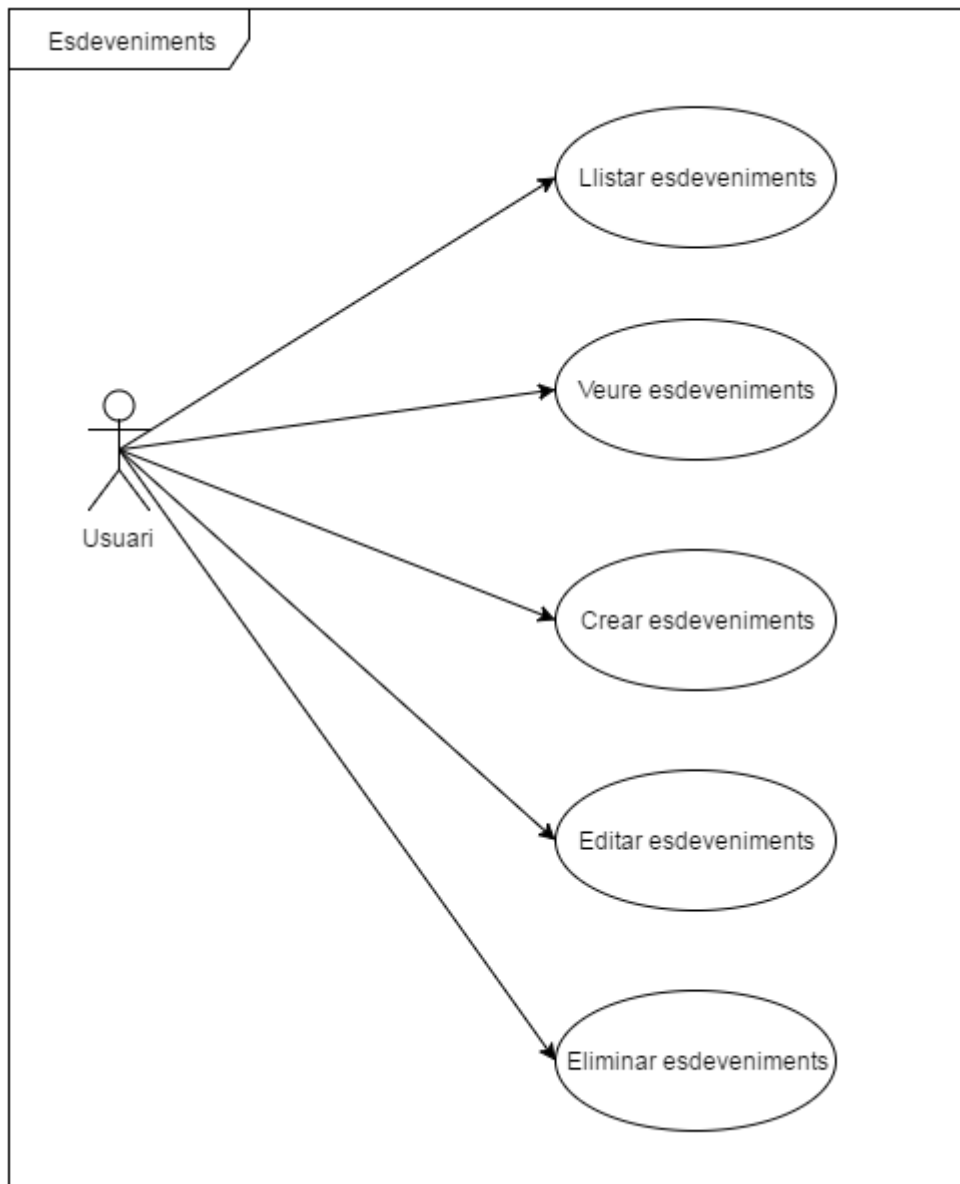
6.3.3 Mur



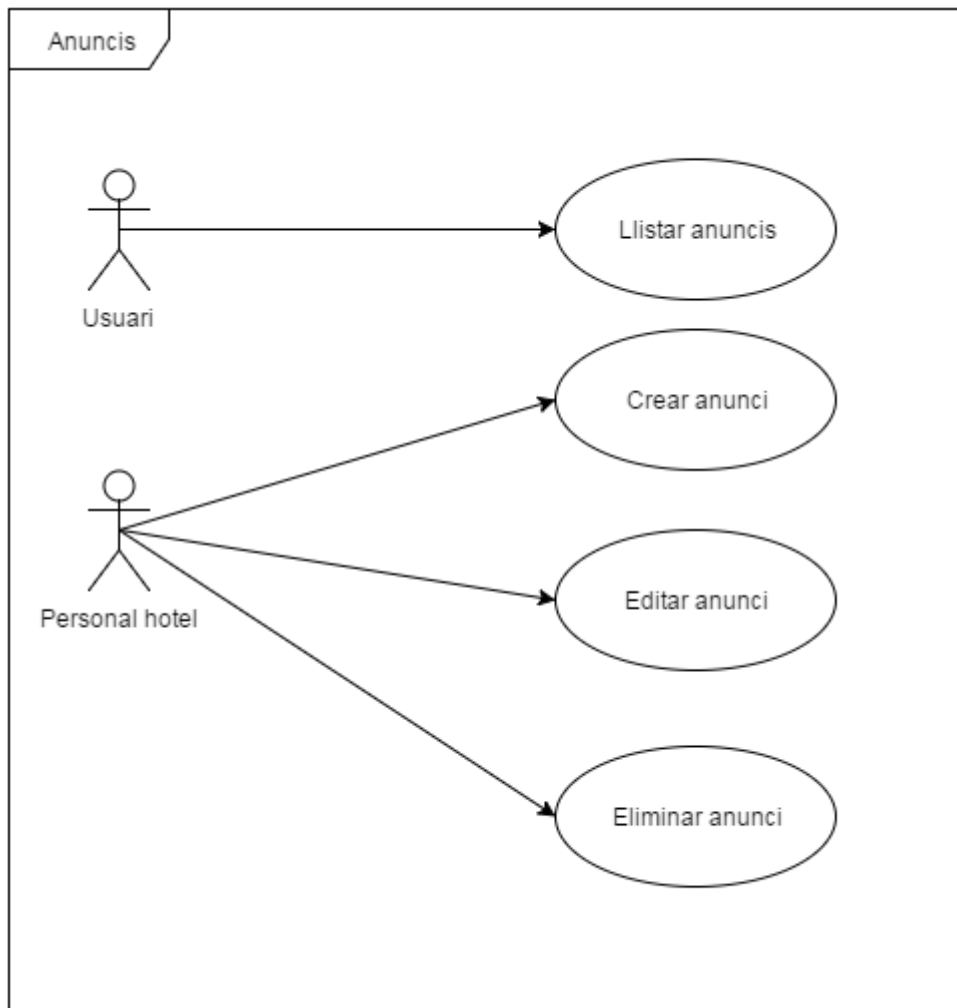
6.3.4 Opinions



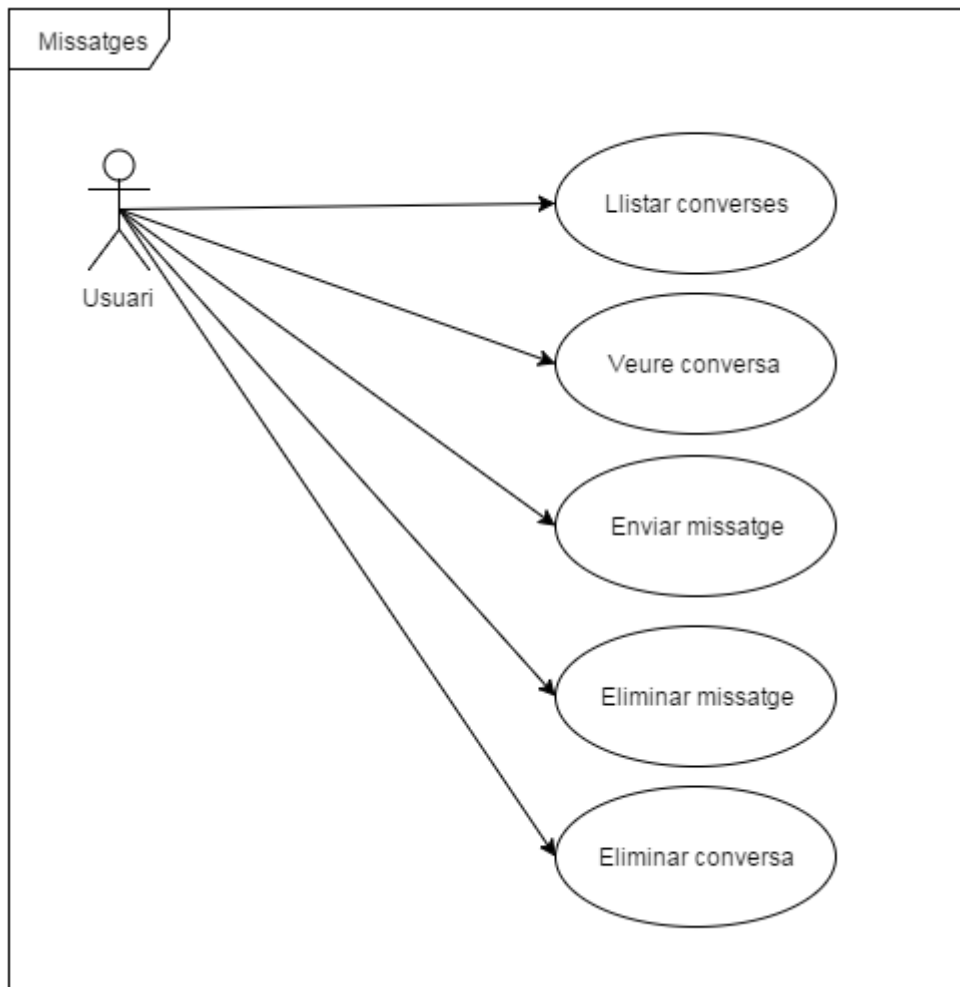
6.3.5 Esdeveniments



6.3.6 Anuncis



6.3.7 Missatges



6.4 Casos d'ús

6.4.1 Usuaris

CU-01	Registre d'usuari		
Precondició	L'usuari no està logat.		
Descripció	Formulari de registre per afegir un nou usuari al sistema.		
Seqüència normal	Pas	Acció	
	1	L'usuari entra a la pàgina amb el formulari de registre.	
	2	L'usuari introdueix les seves dades personals: nom, cognoms, data de naixement, email, contrasenya, confirmació de la contrasenya.	
	3	Fa click al botó "Enviar".	
	4	El sistema valida les dades introduïdes.	
	5	El sistema enregistra l'usuari a la base de dades.	
	6	El sistema envia un mail de confirmació a l'usuari a l'adreça email que ha proporcionat.	
	7	El sistema loga a l'usuari i el redirigeix a la pàgina principal.	
Postcondició	S'ha creat un nou usuari al sistema.		
Excepcions	Pas	Acció	
	1	Un usuari logat accedeix a aquesta pàgina.	
		E.1.1	L'usuari serà redirigit a la pàgina principal.
	4	L'usuari deixa algun dels camps en blanc	
		E.4.1	El sistema informa a l'usuari de quins camps són obligatoris i ha deixat en blanc.
	4	Ja existeix un usuari amb el mail introduït.	
		E.4.2	El sistema informa que ja existeix un usuari amb aquell email.
	4	L'edat de l'usuari és inferior a 18 anys.	
		E.4.3	El sistema informa que només es poden registrar usuaris majors d'edat.
	4	Les dues contrasenyes que ha introduït són diferents	
		E.4.4	El sistema informa que l'usuari ha introduït la contrasenya malament.
Comentaris	Tots els camps del formulari són obligatoris.		

CU-02	Login	
Precondició	L'usuari no està logat.	
Descripció	Formulari per accedir al sistema amb les credencials d'usuari.	
Seqüència normal	Pas	Acció
	1	L'usuari entra a la pàgina amb el formulari de login.
	2	L'usuari introdueix les seves credencials (email i contrasenya).
	3	Fa click al botó entrar.
	4	El sistema valida les dades introduïdes.
	5	El sistema loga a l'usuari i el redirigeix a la pàgina principal.
Postcondició	L'usuari està logat i pot accedir a les seccions de l'aplicació.	
Excepcions	Pas	Acció
	1	Un usuari logat accedeix a aquesta pàgina.
	E.1.1	L'usuari serà redirigit a la pàgina principal.
	4	L'usuari deixa algun dels camps en blanc
	E.4.1	El sistema informa a l'usuari de quins camps són obligatoris i ha deixat en blanc.
	4	L'usuari introdueix un email que no existeix o una contrasenya que no coincideix amb la d'aquell usuari.
	E.4.2	El sistema informa que alguna de les dades es invàlida.
	4	Les dades d'usuari introduïdes corresponen a un usuari desactivat.
	E.4.3	El sistema informa que aquell usuari s'ha desactivat.
Comentaris	Tots els camps del formulari són obligatoris.	

CU-03	Logout	
Precondició	L'usuari està logat.	
Descripció	Botó per a sortir de la sessió d'usuari actual.	
Seqüència normal	Pas	Acció
	1	L'usuari accedeix al menú de l'aplicació.
	2	L'usuari fa click al botó de "Logout".
	3	El sistema acaba la sessió actual de l'usuari i el redirigeix a la pàgina de login.
Postcondició	L'usuari no està logat i no pot accedir a les seccions de l'aplicació.	

CU-04	Llistar usuaris	
Precondició	L'usuari està logat.	
Descripció	Secció de l'aplicació on es llisten els usuaris filtrats per localització o hotel on estan allotjats.	
Seqüència normal	Pas	Acció
	1	L'usuari accedeix al menú de l'aplicació.
	2	L'usuari fa click a "Usuaris".
	3	El sistema redirigeix l'usuari al llistat d'usuaris.
Postcondició	L'usuari es troba al llistat d'usuaris.	

CU-05	Veure perfil d'usuari		
Precondició	L'usuari està logat.		
Descripció	Pàgina amb informació d'un usuari.		
Seqüència normal	Pas	Acció	
	1	L'usuari fa click sobre un usuari.	
	2	El sistema redirigeix a l'usuari al perfil de l'usuari que ha seleccionat.	
	3	Es mostra la informació de l'usuari seleccionat.	
Excepcions	Pas	Acció	
	3	L'usuari accedeix a una fitxa d'un usuari eliminat o desactivat.	
		E.3.1	El sistema mostra una pantalla informant que aquella pàgina ja no existeix.
	3	L'usuari accedeix a la seva pròpia fitxa.	
		E.3.2	Es mostra un botó per editar el perfil.
Comentaris	A la fitxa es mostra: Nom, edat, sexe, descripció, foto de perfil i posts de l'usuari.		

CU-06	Editar perfil d'usuari.		
Precondició	L'usuari està logat.		
Descripció	Pàgina amb informació del propi usuari on pot editar la seva informació.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix al menú de l'aplicació.	
	2	L'usuari fa click a "El meu perfil".	
	3	L'usuari fa click a "Editar perfil".	
	4	El sistema redirigeix l'usuari a la p`qgina per editar la informació de l'usuari.	
	5	Es mostra un formulari per editar la informació de l'usuari.	
	6	L'usuari edita les dades del seu perfil.	
	7	L'usuari fa click a "Guardar".	
	8	El sistema valida les dades introduïdes.	
	9	El sistema actualitza les dades de l'usuari a la base de dades.	
10	El sistema redirigeix a l'usuari a la pàgina del seu perfil.		
Postcondició	Les dades de l'usuari han estat modificades.		
Excepcions	Pas	Acció	
	8	L'usuari no ha emplenat un dels camps obligatoris.	
	E.8.1	El sistema informa a l'usuari de quins camps són obligatoris.	
Comentaris	L'usuari pot editar els camps de Nom, cognoms, descripció i foto de perfil. Els camps de data de naixement i sexe no es poden editar. Noms i cognoms són camps obligatoris.		

CU-07	Eliminar usuari	
Precondició	L'usuari té permisos d'administració sobre usuaris. L'administrador està logat.	
Descripció	L'administrador elimina un usuari del sistema.	
Seqüència normal	Pas	Acció
	1	L'administrador entra en el panell de control d'usuaris.
	2	Selecciona l'usuari que vol eliminar.
	3	Fa click al botó "Eliminar".
	4	El sistema elimina el registre de l'usuari de la base de dades.
	5	El sistema redirigeix a l'administrador al panell de control d'usuaris.
Postcondició	L'usuari seleccionat queda eliminat de la base de dades.	

6.4.2 Hotels

CU-08	Crear hotel		
Precondició	L'usuari té permisos d'administració sobre hotels. L'administrador està logat.		
Descripció	Formulari per afegir un nou hotel al sistema.		
Seqüència normal	Pas	Acció	
	1	L'administrador entra en el panell de control d'hotels.	
	2	Fa click al botó “Crear”.	
	3	El sistema redirigeix a l'administrador al formulari de creació d'hotels.	
	4	L'administrador introdueix les dades de l'hotel.	
	5	Fa click al botó “Guardar”.	
	6	El sistema valida les dades introduïdes.	
	7	El sistema enregistra l'hotel a la base de dades.	
	8	El sistema redirigeix a l'administrador al panell de control d'hotels.	
Postcondició	S'ha creat un nou hotel en el sistema.		
Excepcions	Pas	Acció	
	6	L'administrador deixa algun dels camps obligatoris en blanc.	
		E.6.1	El sistema mostra un missatge de quins camps obligatoris s'ha deixat en blanc.
	6	L'administrador introdueix les dades d'un hotel que ja existeix a la base de dades.	
		E.6.2	El sistema mostra un missatge informant de que ja existeix un hotel amb aquelles dades.
Comentaris	Camps obligatoris: Nom, qualificació, adreça. Camps opcionals: Fotos, descripció.		

CU-09	Llistar hotels	
Precondició	L'usuari està logat.	
Descripció	Pàgina on es llisten els hotels del sistema.	
Seqüència normal	Pas	Acció
	1	L'usuari accedeix al menú de l'aplicació.
	2	Fa click al botó "Hotels".
	3	El sistema redirigeix a l'usuari al llistat d'hotels.
Postcondició	L'usuari es troba al llistat d'hotels.	

CU-10	Fitxa hotel	
Precondició	L'usuari està logat. Com a mínim hi ha un hotel al sistema.	
Descripció	Pàgina de l'aplicació que mostra la informació d'un dels hotels.	
Seqüència normal	Pas	Acció
	1	L'usuari fa click sobre el nom d'un hotel.
	2	El sistema redirigeix a l'usuari a la fitxa de l'hotel seleccionat.
	3	Es mostra la informació de l'hotel seleccionat.
Postcondició	L'usuari es troba en la fitxa de l'hotel.	
Excepcions	Pas	Acció
	2	L'usuari accedeix a una fitxa d'un hotel que no es troba a la base de dades.
	E.2.1	El sistema mostra un missatge dient que aquest hotel no es troba a la base de dades.
Comentaris	A la fitxa de l'hotel es mostra: Nom, qualificació, adreça, mapa de la zona, usuaris allotjats, opinions dels usuaris i fotografies de l'hotel.	

CU-11	Editar informació d'un hotel.		
Precondició	L'usuari té permisos d'administració sobre hotels. L'administrador està logat.		
Descripció	Pàgina amb informació de l'hotel on es pot editar la seva informació.		
Seqüència normal	Pas	Acció	
	1	L'administrador entra en el panell de control d'hotels.	
	2	L'administrador selecciona un dels hotels del llistat.	
	3	L'administrador fa click a "Editar".	
	4	El sistema redirigeix l'usuari a la pàgina per editar la informació de l'hotel.	
	5	Es mostra un formulari per editar la informació de l'hotel.	
	6	L'administrador edita les dades.	
	7	L'administrador fa click a "Guardar".	
	8	El sistema valida les dades introduïdes.	
	9	El sistema actualitza les dades a la base de dades.	
	10	El sistema redirigeix a l'administrador al panell de control d'hotels.	
Postcondició	S'han modificat les dades de l'hotel seleccionat.		
Excepcions	Pas	Acció	
	8	L'usuari no ha emplenat un dels camps obligatoris.	
	E.8.1	El sistema informa a l'usuari de quins camps són obligatoris.	
Comentaris	L'usuari pot editar els camps de Nom, qualificació, descripció, fotografies, adreça. Noms, qualificació i adreça són camps obligatoris.		

CU-12	Eliminar hotel	
Precondició	L'usuari té permisos d'administració sobre hotels. L'administrador està logat.	
Descripció	L'administrador elimina l'hotel del sistema.	
Seqüència normal	Pas	Acció
	1	L'administrador entra en el panell de control d'hotels.
	2	Selecciona l'hotel que vol eliminar.
	3	Fa click al botó "Eliminar".
	4	El sistema elimina el registre de l'hotel de la base de dades.
	5	El sistema redirigeix a l'administrador al panell de control d'hotels.
Postcondició	L'hotel seleccionat queda eliminat de la base de dades.	

6.4.3 Mur

CU-13	Llistar posts	
Precondició	L'usuari està logat.	
Descripció	Pàgina on es llisten els posts que han deixat els usuaris a un hotel determinat.	
Seqüència normal	Pas	Acció
	1	L'usuari accedeix al menú de l'aplicació.
	2	Fa click al botó "Inici".
	3	El sistema redirigeix a l'usuari al mur de l'hotel.
Postcondició	L'usuari es troba al mur de l'hotel.	

CU-14	Crear post		
Precondició	L'usuari està logat.		
Descripció	L'usuari crea un post i s'afegeix com a primer post al mur de l'hotel.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix al mur de l'hotel.	
	2	Emplena el formulari del post.	
	3	L'usuari fa click a "Enviar".	
	4	El sistema valida les dades introduïdes.	
	6	El sistema enregistra el post a la base de dades.	
	5	Es mostra el post al principi del mur de l'hotel.	
Postcondició	S'ha afegit un post al mur de l'hotel.		
Excepcions	Pas	Acció	
	2	L'usuari no ha introduït res en el camp de text.	
	E.2.1	El botó "Enviar" estarà desactivat.	

CU-15	Editar post		
Precondició	L'usuari està logat i ha escrit algun post.		
Descripció	L'usuari edita un dels posts que ha creat en el mur de l'hotel.		
Seqüència normal	Pas	Acció	
	1	L'usuari selecciona un dels seus posts.	
	2	L'usuari fa click al botó "Editar".	
	3	El sistema redirigeix a l'usuari al formulari d'edició de post.	
	4	L'usuari edita el seu post.	
	5	L'usuari fa click a "Guardar".	
	6	El sistema valida les dades introduïdes.	
	7	El sistema actualitza el post a la base de dades.	
	8	El sistema redirigeix a l'usuari al mur de l'hotel.	
Postcondició	S'ha modificat el post de l'usuari amb els canvis que ha introduït.		
Excepcions	Pas	Acció	
	3	L'usuari accedeix a al formulari d'edició d'un post que no és seu.	
		E.3.1	El sistema redirigeix a l'usuari al mur de l'hotel.
	6	L'usuari introdueix dades incorrectes.	
		E.6.1	El sistema mostra un missatge a l'usuari informant de l'error.

CU-16	Eliminar post	
Precondició	L'usuari està logat. L'usuari té com a mínim un post.	
Descripció	L'usuari elimina un dels seus posts del mur de l'hotel.	
Seqüència normal	Pas	Acció
	1	L'usuari selecciona un dels seus posts.
	2	L'usuari fa click al botó "Eliminar".
	3	El sistema redirigeix a l'usuari a la pantalla de confirmació d'eliminació del seu post.
	4	L'usuari fa click a "Acceptar".
	5	El sistema elimina l'esdeveniment de la base de dades.
	6	El sistema redirigeix a l'usuari al mur de l'hotel.
Postcondició	S'ha eliminat el post seleccionat per l'usuari.	
Excepcions	Pas	Acció
	3	L'usuari accedeix a la pantalla de confirmació d'eliminació d'un post que no és seu.
	E.3.1	El sistema mostra un missatge informant a l'usuari que no té permisos per accedir en aquella pàgina.
	4	L'usuari decideix no eliminar el post i fa click al botó "Cancel·lar".
	E.4.1	El sistema redirigeix a l'usuari al mur de l'hotel.

CU-17	Comentar post		
Precondició	L'usuari està logat. Existeix com a mínim un post accessible per l'usuari.		
Descripció	L'usuari pot afegir un comentari a un post del mur de l'hotel.		
Seqüència normal	Pas	Acció	
	1	L'usuari entra al mur de l'hotel.	
	2	L'usuari selecciona el post que vol comentar.	
	3	L'usuari fa click al botó "Comentar".	
	4	Apareix un formulari per introduir-hi el comentari.	
	5	L'usuari introdueix el comentari i fa click a "Enviar".	
	6	El sistema valida les dades introduïdes.	
	7	L'usuari enregistra el comentari a la base de dades.	
	8	El sistema afegeix el comentari a la llista de comentaris del post.	
Postcondició	S'ha creat un comentari i s'ha afegit a la llista de comentaris del post seleccionat.		
Excepcions	Pas	Acció	
	5	L'usuari ha introduït un comentari buit.	
	E.5.1	El botó "Enviar" està desactivat.	

CU-18	Editar comentari		
Precondició	L'usuari està logat. L'usuari ha escrit algun comentari a un post.		
Descripció	Acció per poder editar un dels comentaris realitzats per l'usuari.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix al mur de l'hotel.	
	2	L'usuari selecciona el comentari que vol editar.	
	3	L'usuari fa click al botó “Editar”.	
	4	El sistema redirigeix l'usuari a la pàgina d'edició del comentari.	
	5	L'usuari modifica el seu comentari i fa click en “Guardar”.	
	6	El sistema valida les dades introduïdes per l'usuari.	
	7	El sistema actualitza el comentari a la base de dades.	
	8	El sistema redirigeix a l'usuari al mur de l'hotel.	
Postcondició	S'ha modificat el comentari de l'usuari.		
Excepcions	Pas	Acció	
	4	Un usuari accedeix a la pàgina d'edició d'un comentari que no és seu.	
		E.4.1	El sistema mostra un missatge informant a l'usuari que no té permisos per accedir en aquella pàgina.
	6	L'usuari ha introduït un comentari buit.	
		E.6.1	El sistema notifica a l'usuari que el comentari no pot estar buit.

CU-19	Eliminar comentari	
Precondició	L'usuari està logat. L'usuari ha escrit algun comentari a un post.	
Descripció	Acció per poder eliminar un dels comentaris realitzats per l'usuari.	
Seqüència normal	Pas	Acció
	1	L'usuari accedeix al mur de l'hotel.
	2	L'usuari selecciona el seu comentari que vol eliminar.
	3	L'usuari fa click al botó "Eliminar".
	4	El sistema redirigeix a l'usuari a la pantalla de confirmació d'eliminació del seu comentari.
	5	L'usuari fa click a "Acceptar".
	6	El sistema elimina l'esdeveniment de la base de dades.
	7	El sistema redirigeix a l'usuari al mur de l'hotel.
Postcondició	S'ha eliminat el comentari seleccionat per l'usuari.	
Excepcions	Pas	Acció
	4	L'usuari accedeix a la pantalla de confirmació d'eliminació d'un post que no és seu.
	E.4.1	El sistema mostra un missatge informant a l'usuari que no té permisos per accedir en aquella pàgina.
	5	L'usuari decideix no eliminar el post i fa click al botó "Cancel·lar".
	E.5.1	El sistema redirigeix a l'usuari al mur de l'hotel.

6.4.4 Opinions

CU-20	Llistar opinions		
Precondició	L'usuari està logat.		
Descripció	Llistar on es mostren les opinions que han realitzat els usuaris d'un hotel.		
Seqüència normal	Pas	Acció	
	1	L'usuari selecciona un hotel.	
	2	L'usuari accedeix a la fitxa de l'hotel.	
	3	L'usuari fa click al botó "Veure opinions".	
	4	El sistema redirigeix a l'usuari al llistat d'opinions de l'hotel.	
Postcondició	L'usuari es troba al llistat d'opinions.		
Excepcions	Pas	Acció	
	4	L'usuari accedeix al llistat d'opinions d'un hotel sense opinions.	
	E.4.1	El sistema informa a l'usuari que aquella pàgina no existeix.	
Comentaris	Les opinions estan ordenades de més noves a més antigues. Es poden filtrar per tipus de viatge (Lleure, familiar, parella, negocis...).		
	En el detall de la opinió hi ha el comentari positiu i negatiu de l'usuari, la resposta de l'hotel i les valoracions de l'usuari de neteja, situació, relació qualitat-preu i servei.		

CU-21	Crear opinió		
Precondició	L'usuari està logat. L'usuari ha estat allotjat a l'hotel.		
Descripció	L'usuari crea una opinió per a un hotel.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix a la fitxa de l'hotel.	
	2	L'usuari fa click al botó “Crear opinió”.	
	3	El sistema redirigeix a l'usuari al formulari de creació d'opinions.	
	4	L'usuari empena la dades del formulari i fa click a “Enviar”.	
	5	El sistema valida les dades introduïdes per l'usuari.	
	6	El sistema calcula la puntuació d'aquella opinió en base a les valoracions de l'usuari.	
	7	El sistema enregistra l'opinió a la base de dades.	
	8	El sistema afegeix aquella valoració a la mitjana de valoracions d'opinions de l'hotel.	
	9	El sistema actualitza les dades de l'hotel a la base de dades.	
	10	El sistema redirigeix a l'usuari a la llista d'opinions de l'hotel.	
Postcondició	S'ha creat una nova opinió per l'hotel seleccionat.		
Excepcions	Pas	Acció	
	3	L'usuari accedeix al formulari de crear opinió d'un hotel en el qual no ha estat allotjat.	
		E.3.1	El sistema informa a l'usuari que no té permisos per accedir en aquella secció.
	5	L'usuari no ha introduït totes les dades obligatòries.	
		E.5.1	El sistema informa de quins camps són els obligatoris.
Comentaris	Els camps del formulari són: Tipus de viatge, comentari positiu i negatiu de l'usuari, neteja, situació,relació qualitat-preu i servei.		

CU-22	Editar opinió	
Precondició	L'usuari ha de tenir permisos d'administrador. L'administrador ha d'estar logat.	
Descripció	L'administrador pot afegir un comentari a la opinió de l'usuari sobre l'hotel.	
Seqüència normal	Pas	Acció
	1	L'administrador accedeix al panell de control d'opinions.
	2	L'administrador selecciona una de les opinions.
	3	El sistema redirigeix a l'administrador al formulari d'edició de opinions.
	4	L'administrador afegeix un comentari a la opinió i fa click a "Guardar".
	5	El sistema actualitza les dades de la opinió a la base de dades.
	6	El sistema redirigeix a l'administrador al llistat d'opinions.
Postcondició	S'ha afegit un comentari a una de les opinions d'un hotel.	
Comentaris	Els usuaris veuran el comentari de l'administrador com a contestació de l'hotel a la seva opinió.	

CU-23	Eliminar opinió		
Precondició	L'usuari té permisos d'administració sobre opinions. L'administrador està logat.		
Descripció	L'administrador elimina una opinió del sistema.		
Seqüència normal	Pas	Acció	
	1	L'administrador entra en el panell de control d'opinions.	
	2	Selecciona la opinió que vol eliminar.	
	3	Fa click al botó "Eliminar".	
	4	El sistema redirigeix a l'usuari a la pantalla de confirmació d'eliminació d'opinió.	
	5	L'usuari fa click a "Acceptar".	
	6	El sistema elimina el registre de la opinió de la base de dades.	
	7	El sistema redirigeix a l'administrador al panell de control d'opinions.	
Postcondició	La opinió seleccionada queda eliminada de la base de dades.		
Excepcions	Pas	Acció	
	5	L'usuari decideix no eliminar l'esdeveniment i fa click al botó "Cancel·lar".	
		E.5.1	El sistema redirigeix a l'usuari al panell de control d'opinions.

6.4.5 Esdeveniments

CU-24	Llistar esdeveniments		
Precondició	L'usuari està logat.		
Descripció	Llista d'esdeveniments que es realitzen a una localització.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix al menú de l'aplicació.	
	2	L'usuari fa click a "Esdeveniments".	
	3	El sistema redirigeix a l'usuari al llistat d'esdeveniments.	
Postcondició	L'usuari es troba al llistat d'esdeveniments.		
Excepcions	Pas	Acció	
	3	L'usuari accedeix a un llistat d'esdeveniments d'una localització sense esdeveniments.	
	E.3.1	El sistema informa a l'usuari de que aquella pàgina no existeix.	

CU-25	Veure esdeveniment		
Precondició	L'usuari està logat.		
Descripció	Vista del detall d'un esdeveniment.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix al llistat d'esdeveniments.	
	2	L'usuari selecciona un dels esdeveniments.	
	3	El sistema redirigeix a l'usuari al detall de l'esdeveniment seleccionat.	
Postcondició	L'usuari es troba a la pantalla de detall d'un esdeveniment.		
Excepcions	Pas	Acció	
	3	L'usuari accedeix a una pàgina d'un esdeveniment que no existeix.	
		E.3.1	El sistema informa a l'usuari de que aquella pàgina no existeix.
Comentaris	Un esdeveniment mostra el nom, data, descripció, fotografies, usuaris que hi assisteixen i comentaris dels usuaris.		

CU-26	Crear esdeveniment		
Precondició	L'usuari està logat.		
Descripció	Formulari per a la creació d'un esdeveniment.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix al llistat d'esdeveniments.	
	2	L'usuari fa click al botó "Crear esdeveniment".	
	3	El sistema redirigeix a l'usuari al formulari de creació d'esdeveniments.	
	4	L'usuari emplena la dades del formulari i fa click a "Enviar".	
	5	El sistema valida les dades introduïdes per l'usuari.	
	6	El sistema crea l'esdeveniment a la base de dades.	
	7	El sistema redirigeix a l'usuari a la llista d'esdeveniments.	
Postcondició	S'ha creat un esdeveniment.		
Excepcions	Pas	Acció	
	3	L'usuari no ha introduït totes les dades obligatòries.	
	E.3.1	El sistema informa quins camps són obligatoris.	
Comentaris	Els camps del formulari són: Títol, data, descripció. Com a camps opcionals pot introduir una imatge.		

CU-27	Editar esdeveniment		
Precondició	L'usuari està logat. Existeix un esdeveniment creat per l'usuari.		
Descripció	Formulari d'edició d'un esdeveniment.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix a un dels seus esdeveniments.	
	2	L'usuari fa click a "Editar".	
	3	El sistema redirigeix a l'usuari al formulari d'edició d'esdeveniments.	
	4	L'usuari modifica les dades del formulari i fa click a "Enviar".	
	5	El sistema valida les dades introduïdes per l'usuari.	
	6	El sistema modifica les dades de l'esdeveniment a la base de dades.	
	7	El sistema redirigeix a l'usuari a la pàgina de l'esdeveniment.	
Postcondició	S'ha editat l'esdeveniment seleccionat.		
Excepcions	Pas	Acció	
	3	L'usuari accedeix al formulari d'edició d'un esdeveniment que no ha creat ell.	
		E.3.1	El sistema informa a l'usuari que no té permisos per accedir a aquella pàgina.
	5	L'usuari no ha introduït totes les dades obligatòries.	
		E.5.1	El sistema informa quins camps són obligatoris.
Comentaris	Els camps del formulari són: Títol, data, descripció. Com a camps opcionals pot introduir una imatge.		

CU-28	Eliminar esdeveniment	
Precondició	L'usuari està logat. L'usuari té com a mínim un esdeveniment.	
Descripció	L'usuari elimina un dels seus esdeveniments.	
Seqüència normal	Pas	Acció
	1	L'usuari accedeix a un dels seus esdeveniments.
	2	L'usuari fa click al botó "Eliminar".
	3	El sistema redirigeix a l'usuari a la pantalla de confirmació d'eliminació de l'esdeveniment.
	4	L'usuari fa click a "Acceptar".
	5	El sistema elimina l'esdeveniment de la base de dades.
	6	El sistema redirigeix a l'usuari a la llista d'esdeveniments.
Postcondició	S'ha eliminat l'esdeveniment seleccionat per l'usuari.	
Excepcions	Pas	Acció
	3	L'usuari accedeix a la pantalla de confirmació d'eliminació d'un esdeveniment que no és seu.
	E.3.1	El sistema mostra un missatge informant a l'usuari que no té permisos per accedir en aquella pàgina.
	4	L'usuari decideix no eliminar l'esdeveniment i fa click al botó "Cancel·lar".
	E.4.1	El sistema redirigeix a l'usuari al llistat d'esdeveniments.

6.4.6 Anuncis

CU-29	Llistar anuncis	
Precondició	L'usuari està logat.	
Descripció	Pàgina on es mostren els anuncis dels comerços disponibles a la zona.	
Seqüència normal	Pas	Acció
	1	L'usuari accedeix al menú de l'aplicació.
	2	L'usuari fa click al botó "Anuncis".
	3	El sistema redirigeix a l'usuari al llistat d'anuncis.
Postcondició	L'usuari es troba al llistat d'anuncis.	

CU-30	Crear anunci		
Precondició	L'usuari ha de tenir permisos d'administrador. L'administrador ha d'estar logat.		
Descripció	Formulari per a la creació d'un anunci.		
Seqüència normal	Pas	Acció	
	1	L'administrador accedeix al panell de control d'anuncis.	
	2	L'administrador fa click a "Crear anunci".	
	3	El sistema redirigeix a l'administrador al formulari de creació d'un anunci.	
	4	L'administrador emplena la dades del formulari i fa click a "Enviar".	
	5	El sistema valida les dades introduïdes per l'administrador.	
	6	El sistema crea l'anunci a la base de dades.	
	7	El sistema redirigeix a l'administrador al panell de control d'anuncis.	
Postcondició	S'ha creat un anunci nou.		
Excepcions	Pas	Acció	
	5	L'usuari no ha introduït totes les dades obligatòries.	
	E.5.1	El sistema informa de quins camps són els obligatoris.	
Comentaris	Els camps obligatoris són títol i imatge.		

CU-31	Editar anunci		
Precondició	L'usuari ha de tenir permisos d'administrador. L'administrador ha d'estar logat.		
Descripció	Formulari per a l'edició d'un anunci.		
Seqüència normal	Pas	Acció	
	1	L'administrador accedeix al panell de control d'anuncis.	
	2	L'administrador selecciona un anunci i fa click a "Editar".	
	3	El sistema redirigeix a l'administrador al formulari d'edició d'un anunci.	
	4	L'administrador emplena les dades del formulari i fa click a "Guardar".	
	5	El sistema valida les dades introduïdes per l'administrador.	
	6	El sistema modifica l'anunci a la base de dades.	
	7	El sistema redirigeix a l'administrador al panell de control d'anuncis.	
Postcondició	S'ha modificat l'anunci seleccionat.		
Excepcions	Pas	Acció	
	5	L'usuari no ha introduït totes les dades obligatòries.	
	E.5.1	El sistema informa quins camps són obligatoris.	
Comentaris	Els camps obligatoris són títol i imatge.		

CU-32	Eliminar anunci	
Precondició	L'usuari té permisos d'administració sobre anuncis. L'administrador està logat.	
Descripció	L'administrador elimina un anunci del sistema.	
Seqüència normal	Pas	Acció
	1	L'administrador entra en el panell de control d'anuncis.
	2	Selecciona l'anunci que vol eliminar.
	3	Fa click al botó "Eliminar".
	4	El sistema redirigeix a l'usuari a la pantalla de confirmació d'eliminació d'anunci.
	5	L'usuari fa click a "Acceptar".
	6	El sistema elimina el registre de l'anunci de la base de dades.
	7	El sistema redirigeix a l'administrador al panell de control d'anuncis.
Postcondició	L'anunci seleccionat queda eliminada de la base de dades.	
Excepcions	Pas	Acció
	4	L'usuari accedeix a la pantalla de confirmació d'eliminació d'un anunci que no és seu.
	E.4.1	El sistema mostra un missatge informant a l'usuari que no té permisos per accedir a aquella pàgina.
	5	L'usuari decideix no eliminar l'anunci i fa click al botó "Cancel·lar"
	E.5.1	El sistema redirigeix a l'usuari al panell de control d'anuncis..

6.4.7 Missatges

CU-33	Llistar converses		
Precondició	L'usuari està logat.		
Descripció	Pàgina on es llisten les converses que l'usuari té amb altres usuaris del sistema.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix al menú de l'aplicació.	
	2	L'usuari fa click al botó "Missatges".	
	3	El sistema redirigeix a l'usuari al llistat de converses de l'usuari.	
Postcondició	L'usuari es troba en el llistat de converses.		
Excepcions	Pas	Acció	
	3	L'usuari no ha iniciat cap conversa amb altres usuaris.	
	E.3.1	El sistema mostra el llistat buit amb un enllaç al llistat d'usuaris.	
Comentaris	Es llisten les converses amb ordre cronològic de missatge, de més nou a més vell (enviat o rebut). Si alguna de les converses té algun missatge sense llegir es marca de manera diferent.		

CU-34	Veure conversa	
Precondició	L'usuari està logat.	
Descripció	Seqüència de missatges que s'han intercanviat un usuari amb un altre del sistema.	
Seqüència normal	Pas	Acció
	1	L'usuari accedeix al llistat de converses.
	2	L'usuari selecciona una de les converses.
	3	El sistema redirigeix a l'usuari a la vista de la conversa.
Postcondició	L'usuari es troba a la vista de la conversa seleccionada.	
Excepcions	Pas	Acció
	2	L'usuari pot iniciar o veure una conversa amb un altre usuari des del perfil de l'altre usuari fent click al botó "Enviar missatge".
Comentaris	Es mostraran els darrers missatges intercanviats, els més nous a la part de baix. A l'esquerra es mostraran els missatges rebuts i a la dreta els missatges enviats. A baix de tot apareix el formulari d'enviament de nou missatge.	

CU-35	Enviar missatge		
Precondició	L'usuari està logat.		
Descripció	Formulari per a l'enviament d'un nou missatge a un altre usuari del sistema.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix a una conversa amb un altre usuari.	
	2	Es mostren els missatges intercanviats i el formulari d'enviament de nou missatge.	
	3	L'usuari emplena el formulari i fa click a "Enviar".	
	4	El sistema valida les dades introduïdes per l'usuari.	
	5	El missatge apareix al final de la conversa actual.	
	6	El sistema guarda el missatge a la base de dades i el marca com a no llegit pel receptor.	
	7	El sistema envia una notificació a l'usuari receptor avisant-lo que té un missatge nou.	
Postcondició	S'ha enviat un missatge nou a un altre usuari de la plataforma.		
Excepcions	Pas	Acció	
	3	Si l'usuari no ha introduït text al formulari.	
	E.3.1	El botó "Enviar" no farà cap acció.	

CU-36	Eliminar missatge		
Precondició	L'usuari està logat. L'usuari, com a mínim, té una conversa amb, com a mínim, un missatge enviat o rebut.		
Descripció	Acció per a eliminar un dels missatges de la conversa amb un altre usuari.		
Seqüència normal	Pas	Acció	
	1	L'usuari accedeix a una de les converses.	
	2	L'usuari selecciona un dels missatges de la conversa.	
	3	L'usuari fa click al botó de “Eliminar”.	
	4	El sistema redirigeix a l'usuari a la pantalla de confirmació d'eliminació de missatge.	
	5	L'usuari fa click a “Acceptar”.	
	6	El sistema marca com eliminat el missatge per a aquest usuari.	
	7	El sistema redirigeix a l'usuari a la conversa.	
Postcondició	S'ha eliminat el missatge seleccionat per l'usuari emissor.		
Excepcions	Pas	Acció	
	4	L'usuari accedeix a la pantalla de confirmació d'eliminació d'un missatge que no és seu.	
		E.4.1	El sistema mostra un missatge informant a l'usuari que no té permisos per accedir a aquella pàgina.
	5	L'usuari decideix no eliminar el missatge i fa click al botó “Cancel·lar”.	
		E.5.1	El sistema redirigeix a l'usuari a la conversa.
Comentari	Si un usuari elimina un missatge no s'ha d'eliminar pel receptor.		

CU-37	Eliminar conversa	
Precondició	L'usuari està logat. L'usuari, com a mínim, té una conversa.	
Descripció	Acció per a eliminar una de les converses amb un altre usuari.	
Seqüència normal	Pas	Acció
	1	L'usuari accedeix al llistat de converses.
	2	L'usuari selecciona una de les converses.
	3	L'usuari fa click al botó de "Eliminar".
	4	El sistema redirigeix a l'usuari a la pantalla de confirmació d'eliminació de conversa.
	5	L'usuari fa click a "Acceptar".
	6	El sistema marca com a eliminada la conversa per a aquest usuari.
	7	El sistema redirigeix a l'usuari al llistat de converses.
Postcondició	S'ha eliminat la conversa seleccionada per l'usuari emissor.	
Excepcions	Pas	Acció
	4	L'usuari accedeix a la pantalla de confirmació d'eliminació de conversa que no és seu.
	E.4.1	El sistema mostra un missatge informant a l'usuari que no té permisos per accedir en aquella pàgina.
	5	L'usuari decideix no eliminar la conversa i fa click al botó "Cancel·lar".
	E.5.1	El sistema redirigeix a l'usuari al llistat de converses.
Comentaris	Si un usuari elimina una conversa no s'ha d'eliminar pel receptor.	

6.5 Requisits no funcionals

Requisit no funcional	RNF-01
Tipus	Accessibilitat
Descripció	El sistema no podrà estar inaccessible mentre s'estigui actualitzant.
Criteri de satisfacció	Es realitzaran proves en un entorn el més semblant possible al de producció abans de desplegar una nova actualització.
Prioritat	Alta

Requisit no funcional	RNF-02
Tipus	Accessibilitat
Descripció	El sistema serà accessible des dels navegadors més utilitzats.
Criteri de satisfacció	Es faran proves d'integració utilitzant diferents navegadors.
Prioritat	Alta

Requisit no funcional	RNF-03
Tipus	Capacitat
Descripció	El sistema haurà de tenir capacitat per emmagatzemar dades segons la demanda.
Criteri de satisfacció	Es faran proves d'estrès sobre un entorn de test utilitzant per comprovar que l'auto-escalat d'Amazon Web Services funciona correctament.
Prioritat	Alta

Requisit no funcional	RNF-04
Tipus	Capacitat de prova
Descripció	El sistema ha de ser capaç de ser testejat a tots els nivells: Visual, usabilitat, accessibilitat, codi i rendiment.
Criteri de satisfacció	Es realitzaran test d'integració, unitaris, proves d'estes i proves amb usuaris reals.
Prioritat	Mitja

Requisit no funcional	RNF-05
Tipus	Desplegament
Descripció	El sistema ha de tenir un procediment documentat de desplegament de l'aplicació tant per a servidors de desenvolupament com de producció.
Criteri de satisfacció	S'entregarà conjuntament amb el codi un sèrie d'scripts per a realitzar desplegament automatitzat.
Prioritat	Alta

Requisit no funcional	RNF-06
Tipus	Disponibilitat
Descripció	El sistema estarà disponible 24 hores al dia els 365 dies de l'any.
Criteri de satisfacció	El sistema ha de tenir una taxa de funcionament superior al 95%.
Prioritat	Alta

Requisit no funcional	RNF-07
Tipus	Documentació
Descripció	El sistema ha d'estar documentant en tot moment. Tant les funcionalitats, el codi, i els processos que impliquin el seu manteniment i desplegament a servidors.
Criteri de satisfacció	Juntament amb el codi font s'entregarà la documentació pertinent.
Prioritat	Alta

Requisit no funcional	RNF-08
Tipus	Escalabilitat
Descripció	El sistema haurà de ser capaç d'atendre sol·licituds segons la demanda.
Criteri de satisfacció	Es realitzaran proves d'estrès per verificar l'escalabilitat del sistema.
Prioritat	Alta

Requisit no funcional	RNF-09
Tipus	Eficiència
Descripció	El sistema ha d'estar dissenyat per a que consumeixi el mínim possible de dades, recursos i temps.
Criteri de satisfacció	Es farà ús de sistemes de caché per a minimitzar temps de càrrega i accessos innecessaris a disc.
Prioritat	Mitja

Requisit no funcional	RNF-10
Tipus	Integritat
Descripció	El sistema realitzarà còpies de seguretat constantment.
Criteri de satisfacció	Es realitzarà un pla de còpies de seguretat.
Prioritat	Alta

Requisit no funcional	RNF-11
Tipus	Integritat
Descripció	El sistema disposarà d'un sistema de logs.
Criteri de satisfacció	Un cop desplegada l'aplicació es podran consultar els logs de sistema que ha enregistrat el servidor.
Prioritat	Alta

Requisit no funcional	RNF-12
Tipus	Mantenibilitat
Descripció	El sistema ha de ser flexible a futurs canvis i apte per a la incorporació de noves funcionalitats.
Criteri de satisfacció	Es realitzarà un codi estructurat i documentat per facilitar futurs canvis.
Prioritat	Alta

Requisit no funcional	RNF-13
Tipus	Privacitat
Descripció	El sistema informarà als usuaris de la política de privacitat sobre les dades que disposa.
Criteri de satisfacció	L'usuari tindrà a la seva disposició la política de privacitat des de el menú de l'aplicació.
Prioritat	Alta

Requisit no funcional	RNF-14
Tipus	Privacitat
Descripció	Totes les dades seran tractades segons estableix la LOPD i la directiva de protecció de dades 95/46 de la UE.
Criteri de satisfacció	Es realitzarà un consultoria per a verificar-ho.
Prioritat	Alta

Requisit no funcional	RNF-15
Tipus	Recuperació de desastres
Descripció	El sistema ha de disposar d'un procediment en cas de desastre per a tornar a posar-lo en funcionament el més aviat possible.
Criteri de satisfacció	S'adjuntarà un document amb el protocol a seguir en cas de desastre.
Prioritat	Alta

Requisit no funcional	RNF-16
Tipus	Rendiment
Descripció	Les insercions i modificacions a base de dades tardaran menys d'un minut en condicions normals.
Criteri de satisfacció	Es realitzaran proves d'estrès per a mesurar aquestes mètriques.
Prioritat	Alta

Requisit no funcional	RNF-17
Tipus	Rendiment
Descripció	Les peticions al servidor respondran en menys de 2 segons en condicions normals.
Criteri de satisfacció	Es realitzaran proves d'estrès per a mesurar aquestes mètriques.
Prioritat	Alta

Requisit no funcional	RNF-18
Tipus	Seguretat
Descripció	El sistema serà el més segur possible per evitar atacs informàtics.
Criteri de satisfacció	El framework disposa de sistemes per prevenir la majoria dels atacs. Es realitzarà un consultoria per a verificar-ho.
Prioritat	Alta

Requisit no funcional	RNF-19
Tipus	Seguretat
Descripció	Totes les peticions i respostes del sistema hauran d'estar encriptades.
Criteri de satisfacció	Totes les peticions al sistema estarà sota el protocol HTTPS.
Prioritat	Alta

Requisit no funcional	RNF-21
Tipus	Suport
Descripció	El sistema disposarà d'una secció d'ajuda.
Criteri de satisfacció	L'usuari podrà accedir-hi en tot moment des de el menú de l'aplicació.
Prioritat	Baixa

Requisit no funcional	RNF-22
Tipus	Usabilitat
Descripció	El sistema ha de ser fàcil d'utilitzar.
Criteri de satisfacció	Es realitzaran proves amb usuaris reals i es valorarà la usabilitat de cada funció.
Prioritat	Alta

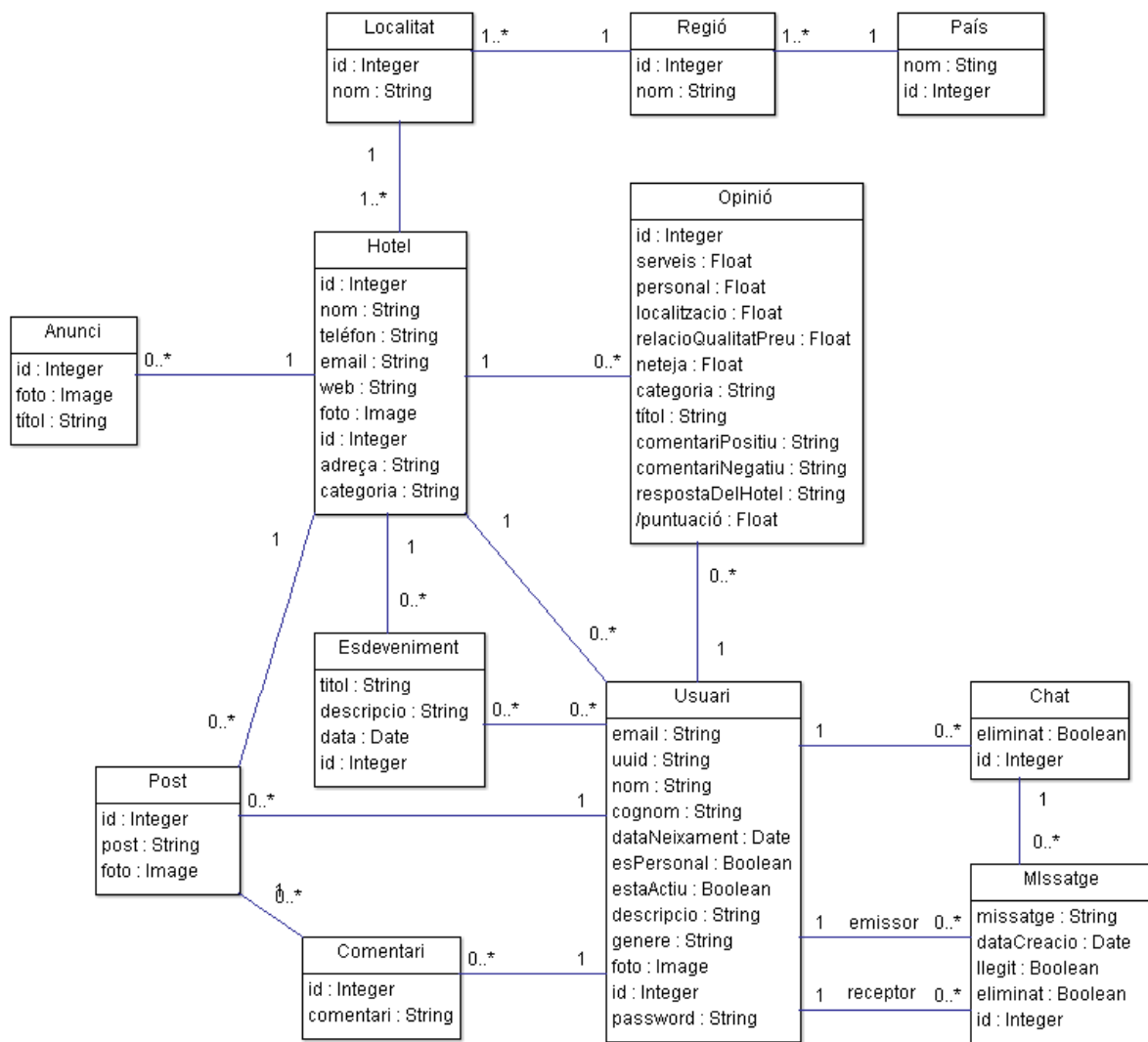
Requisit no funcional	RNF-23
Tipus	Usabilitat
Descripció	El sistema ha d'utilitzar un llenguatge comprensible per a qualsevol usuari.
Criteri de satisfacció	El vocabulari utilitzat està adaptat a l'usuari objectiu i no genera confusions.
Prioritat	Alta

Requisit no funcional	RNF-24
Tipus	Usabilitat
Descripció	Els sistema ha d'utilitzar icones que siguin familiars per a l'usuari.
Criteri de satisfacció	S'utilitzaran icones estàndard o de ús normalitzat.
Prioritat	Alta

Requisit no funcional	RNF-25
Tipus	Usabilitat
Descripció	El sistema s'ha de poder utilitzar fàcilment des de qualsevol dispositiu (sobretaula, portàtil, tauleta o mòbil)
Criteri de satisfacció	Es realitzaran proves d'integració en diferents resolucions i dispositius.
Prioritat	Alta

7 Especificació

7.1 Esquema conceptual



7.1.1 Restriccions de clau externa

- (Usuari: uuid)
- (Hotel: id)
- (Post: id)
- (Comentari: id)
- (Opinió: id)
- (Esdeveniment: id)
- (Anunci: id)
- (País: id)
- (Regió: id)
- (Localitat: id)
- (Chat: id)
- (Missatge: id)

7.1.2 Restriccions d'integritat textuals

- Un usuari només pot estar allotjat a un hotel.
- Un missatge no pot tenir com a emissor i receptor al mateix usuari.
- Un usuari no pot crear una opinió d'un hotel on no ha estat allotjat.

7.1.3 Informació derivada

- $\text{puntua\c{c}i\o} = \frac{\text{neteja} + \text{serveis} + \text{personal} + \text{localitzaci\o} + \text{relacioQualitatPreu}}{5}$

7.1.4 Descripció de les entitats

Usuari

Usuari del sistema.

Atribut	Descripció
email	Correu electrònic de l'usuari necessari per accedir al sistema.
password	Contrasenya necessària per poder accedir al sistema.
nom	Nom del usuari.
cognom	Cognom del usuari.
dataDeNaixament	Data en que va néixer l'usuari.
esPersonal	Indica si l'usuari té privilegis per accedir al panell d'administració.
estaActiu	Indica si l'usuari està activat o no.
descripcio	Breu descripció personal de l'usuari.
genere	Gènere de l'usuari.
foto	Fotografia de perfil de l'usuari.

Hotel

Emmagatzema la informació d'un hotel.

Atribut	Descripció
nom	Nom de l'hotel.
telèfon	Telèfon de contacte de l'hotel.
email	Correu electrònic de contacte de l'hotel.
web	Pàgina web de l'hotel.
adreça	Adreça on es troba l'hotel.
categoria	Categoria de l'hotel.

foto	Fotografia de l'hotel.
-------------	------------------------

Post

Comentaris o fotografies del mur de l'hotel.

Atribut	Descripció
post	Comentari del post afegit per l'usuari.
foto	Fotografia del post afegit per l'usuari.

Comentari

Comentari d'un usuari a un post.

Atribut	Descripció
comentari	Comentari del usuari.

Opinió

Opinió d'un usuari a un hotel.

Atribut	Descripció
serveis	Valoració de l'usuari sobre els serveis de l'hotel.
personal	Valoració de l'usuari sobre el personal de l'hotel.
localització	Valoració de l'usuari sobre la localització de l'hotel.
relacioQualitatPreu	Valoració de l'usuari sobre la relació qualitat preu de l'hotel.
neteja	Valoració de l'usuari sobre la neteja de l'hotel.
categoria	Tipus de viatge que feia l'usuari al visitar l'hotel (amb família, amb parella, de negocis, oci...)
títol	Títol de la opinió.
comentariPositiu	Comentari positiu de l'usuari sobre l'hotel.
comentariNegatiu	Comentari negatiu de l'usuari sobre l'hotel.
respostaDelHotel	Resposta de l'hotel a la opinió de l'usuari.
/puntuació	Puntuació total de l'opinió de l'usuari.

Esdeveniment

Emmagatzema la informació d'un esdeveniment .

Atribut	Descripció
---------	------------

títol	Títol de l'esdeveniment.
descripció	Descripció de l'esdeveniment.
data	Data de quan es realitzarà l'esdeveniment.

Anunci

Emmagatzema la informació d'un anunci.

Atribut	Descripció
títol	Títol de l'anunci.
foto	Fotografia de l'anunci.

País

Emmagatzema la informació d'un país.

Atribut	Descripció
nom	Nom del país.

Regió

Emmagatzema la informació d'una regió.

Atribut	Descripció
nom	Nom de la regió.

Localitat

Emmagatzema la informació d'una localitat.

Atribut	Descripció
nom	Nom de la localitat.

Xat

Emmagatzema la informació d'una conversa entre dos usuaris.

Atribut	Descripció
eliminat	Marca el xat com eliminat perquè no aparegui al llistat de converses.

Missatge

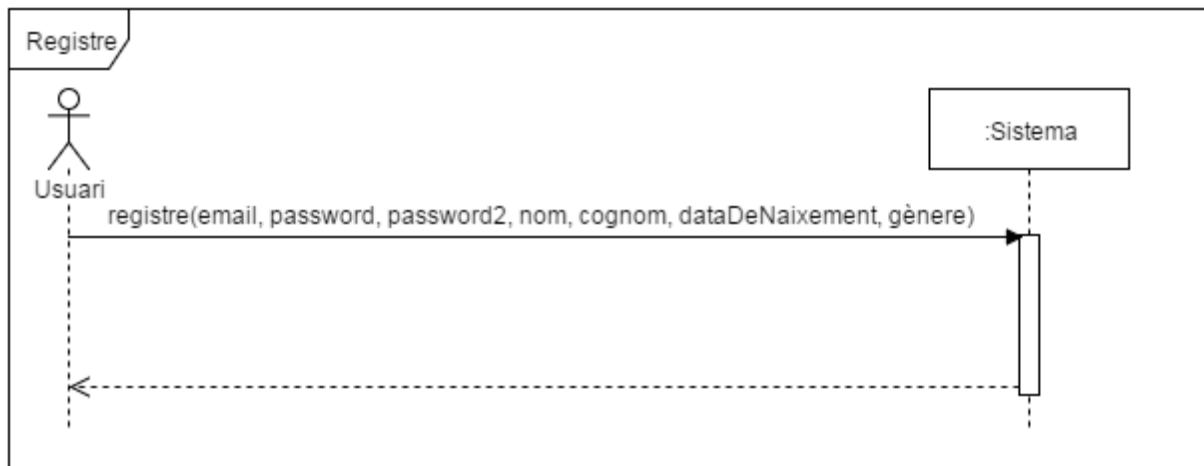
Emmagatzema la informació d'un missatge enviat entre usuaris.

Atribut	Descripció
missatge	Contingut del missatge.
dataCreació	Data en la que es va crear el missatge.
eliminat	Marca si el missatge ha estat eliminat perquè no aparegui a la llista de missatges.
llegit	Marca si el missatge ha estat llegit per l'usuari.

7.2 Esquema del comportament

7.2.1 Usuaris

7.2.1.1 Registre



Context: Sistema::registre(*email*: string, *password*: string, *password2*: string, *nom*: string, *cognom*: string, *dataDeNaixement*: date, *gènere*: int)

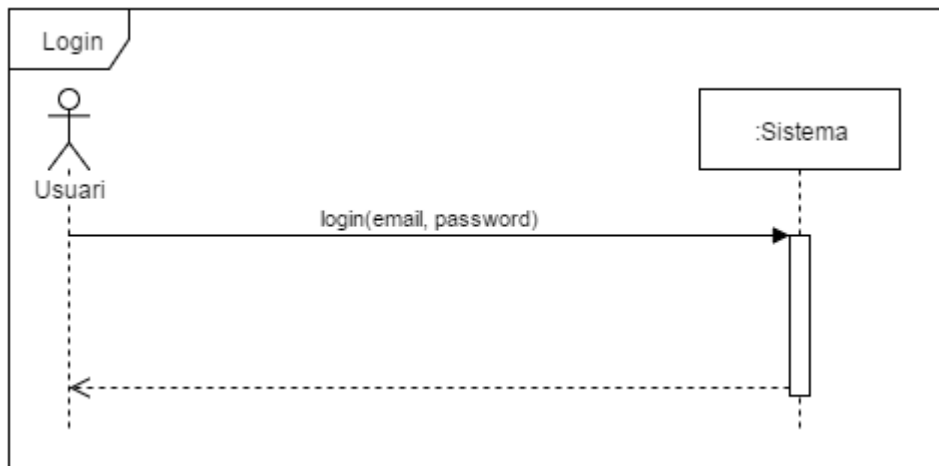
Pre: *email* , *password*, *password2*, *nom*, *cognom*, *dataDeNaixement* i *gènere* no estan buits.

password i *password2* han de ser iguals.

dataDeNeixament ha de ser d'una persona major d'edat.

Post: S'ha creat una nova instància d'usuari amb adreça electrònica igual a *email*, contrasenya igual a *password*, nom igual a *nom*, cognom igual a *cognom*, data de naixement igual a *dataDeNaixement* i gènere igual a *gènere*.

7.2.1.2 Login



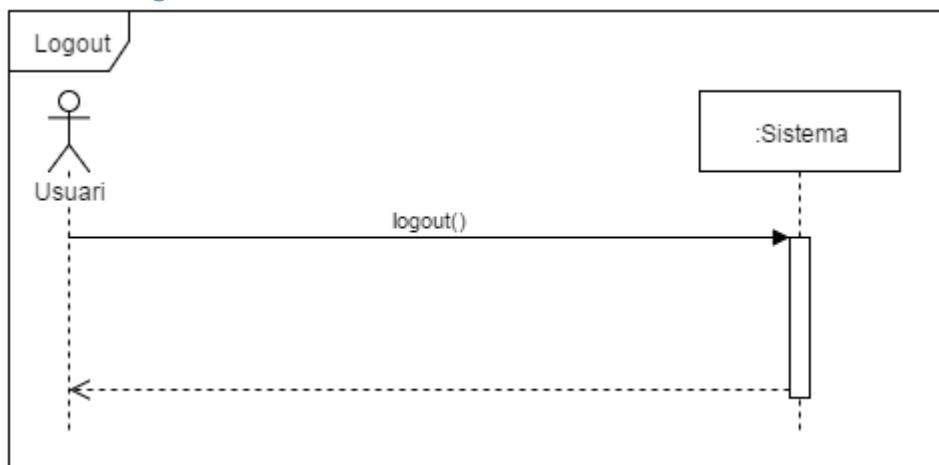
Context: Sistema::login(*email*: string, *password*: string)

Pre: L'usuari no està logat.

email i *password* no estan buits.

Post: Si existeix un usuari amb adreça electrònica *email* i contrasenya *password* l'usuari serà redirigit a la pantalla principal.

7.2.1.3 Logout

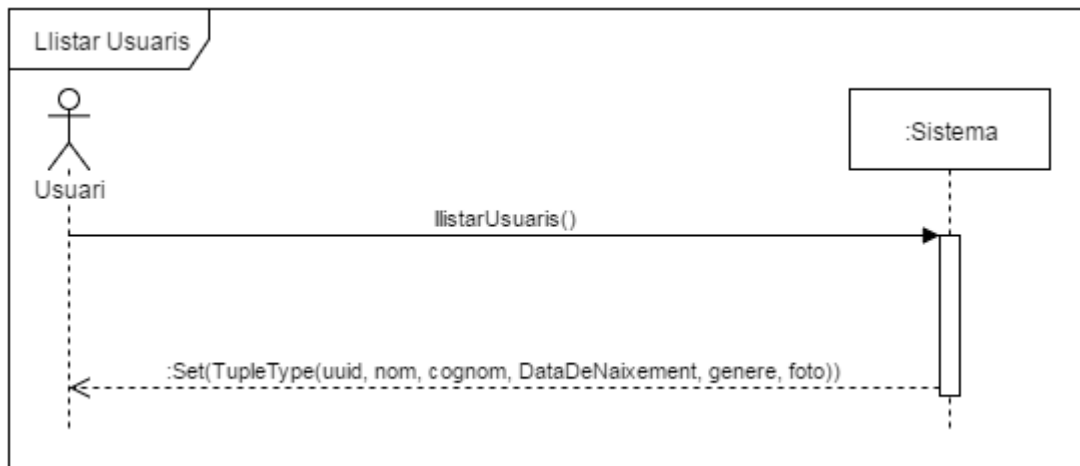


Context: Sistema::logout()

Pre: L'usuari està logat.

Post: Es tanca la sessió de l'usuari i es trobarà a la pantalla de login.

7.2.1.4 Llistar usuaris

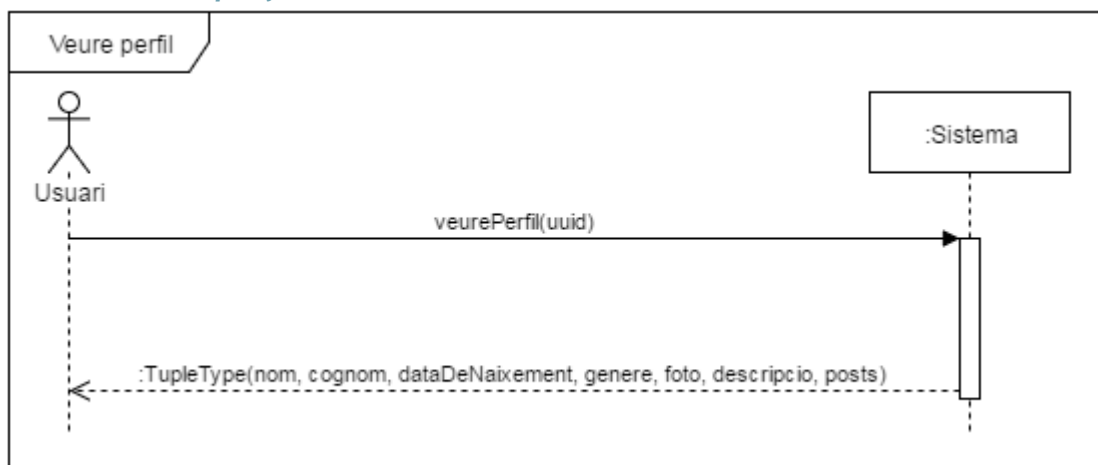


Context: Sistema::llistarUsuaris() :Set(TupleType(uuid: string, nom: string, cognom: string, dataDeNaixement: date, gènere: string, foto: image))

Pre: L'usuari està logat.

Post: L'usuari veu la llista d'usuaris.

7.2.1.5 Veure perfil



Context: Sistema::veurePerfil(uuid: int) :TupleType(nom: string, cognom: string, DataDeNaixement: date, gènere: string, foto: image, descripció: string, posts: Set(post: post))

Pre: L'usuari està logat.

Post: L'usuari veu les dades de l'usuari amb identificador *uuid*.

7.2.1.6 Editar perfil



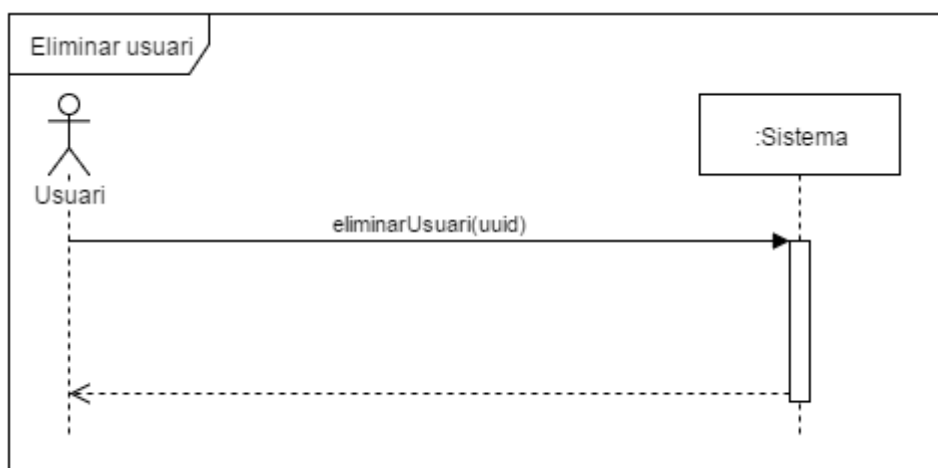
Context: Sistema::editarPerfil(*uuid*: string, *nom*: string, *cognom*: string, *foto*: image, *descripció*: string) :TupleType(*nom*: string, *cognom*: string, *dataDeNaixement*: date, *gènere*: string, *foto*: image, *descripció*: string, *posts*: Set(*post*: post))

Pre: L'usuari està logat.

L'identificador de l'usuari es *uuid*.

Post: La instància d'Usuari *U* identificat per *uuid* té com a nom *nom*, com a cognom *cognom*, la seva imatge de perfil és *foto* i la seva descripció és *descripció*.

7.2.1.7 Eliminar usuari



Context: Sistema::eliminar_usuari(*uuid*: int)

Pre: L'usuari està logat.

Post: S'ha eliminat la instància d'Usuari *U* identificat amb *uuid* del sistema.

7.2.2 Hotel

7.2.2.1 Crear hotel



Context: Sistema::crearHotel(*nom*: string, *categoria*: string, *telèfon*: string, *email*: string, *web*: string: *adreça*: string, *localitat*: localitat)

Pre: *nom*, *category*, *localitat* no estan buits.

Existeix una instància de Locality *localitat*.

Post: S'ha creat una instància d'hotel *H* amb nom igual a *nom*, categoria igual a *categoria*, telèfon igual a *telèfon*, adreça electrònica igual a *email*, adreça igual a *adreça* i pertany a la localitat *localitat*.

7.2.2.2 Llistar hotels

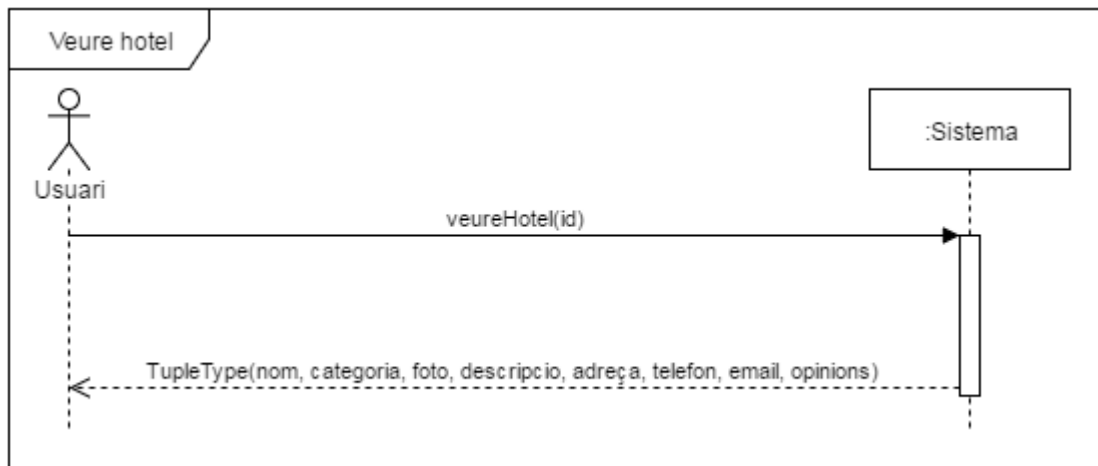


Context: Sistema::llistarHotels() :Set(TupleType(*id*: int, *nom*: string, *categoria*: string, *foto*: image))

Pre: L'usuari està logat.

Post: L'usuari veu la llista d'hotels.

7.2.2.3 Veure hotel

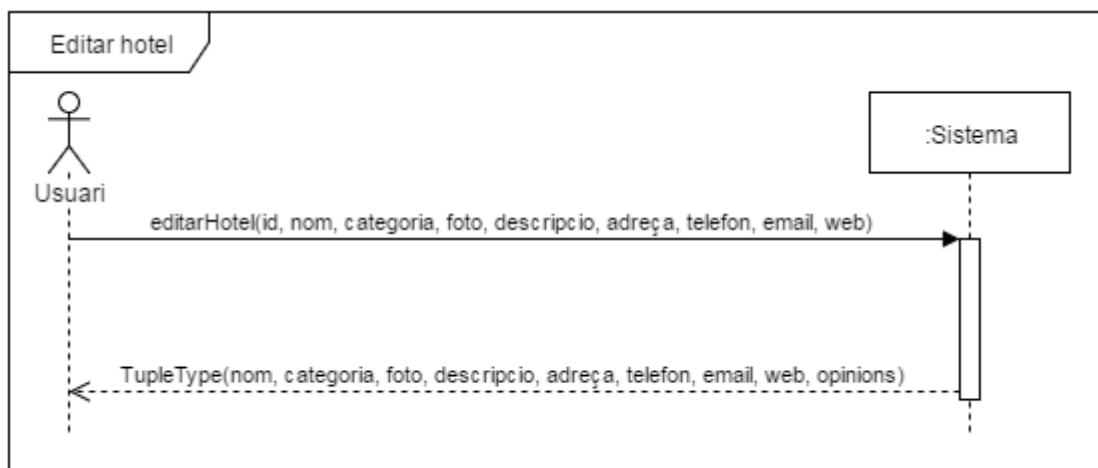


Context: Sistema::veureHotel(id: int) :TupleType(*nom*: string, *categoria*: string, *foto*: image, *descripció*: string, *adreça*: string, *telèfon*: string, *email*: string, *opinions*: Set(*opinió*: opinió))

Pre: L'usuari està logat.

Post: L'usuari veu les dades de l'hotel identificat per *id*.

7.2.2.4 Editar hotel



Context: Sistema::editarHotel(id: int, *nom*: string, *categoria*: string, *foto*: image, *descripció*: string, *adreça*: string, *telèfon*: string, *email*: string, *web*: string) :TupleType(*nom*: string, *categoria*: string, *foto*: image, *descripció*: string, *adreça*: string, *telèfon*: string, *web*: string, *opinions*: Set(*opino*: opinió))

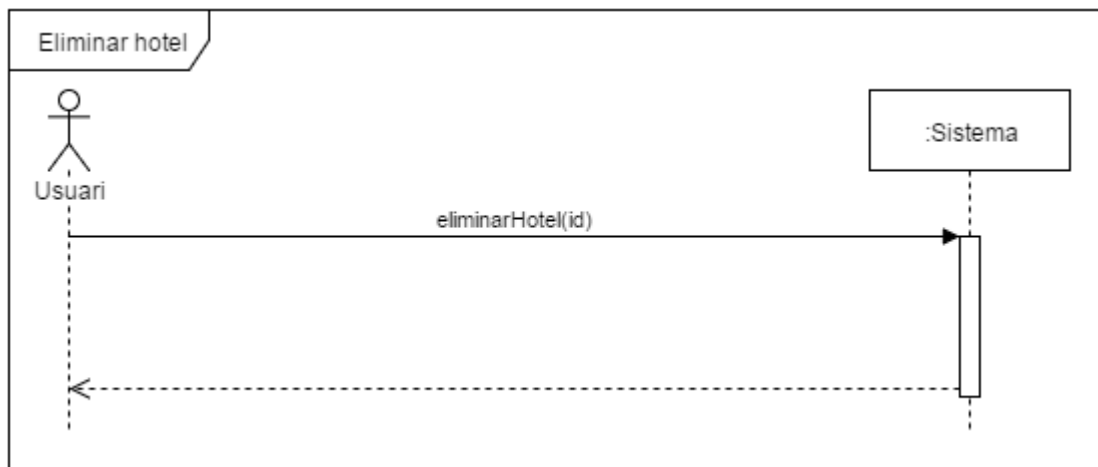
Pre: L'usuari està logat.

Existeix un hotel amb identificador *id*.

Post: La instància d'Hotel *H* identificat per *id* té com a nom *nom*, la seva categoria és *categoria*, la seva imatge de perfil és *foto*, la seva descripció és *descripció*,

la seva adreça és *adreça*, el seu telèfon és *telèfon*, la seva adreça electrònica és *email* i la seva pàgina web és *web*.

7.2.2.5 Eliminar hotel



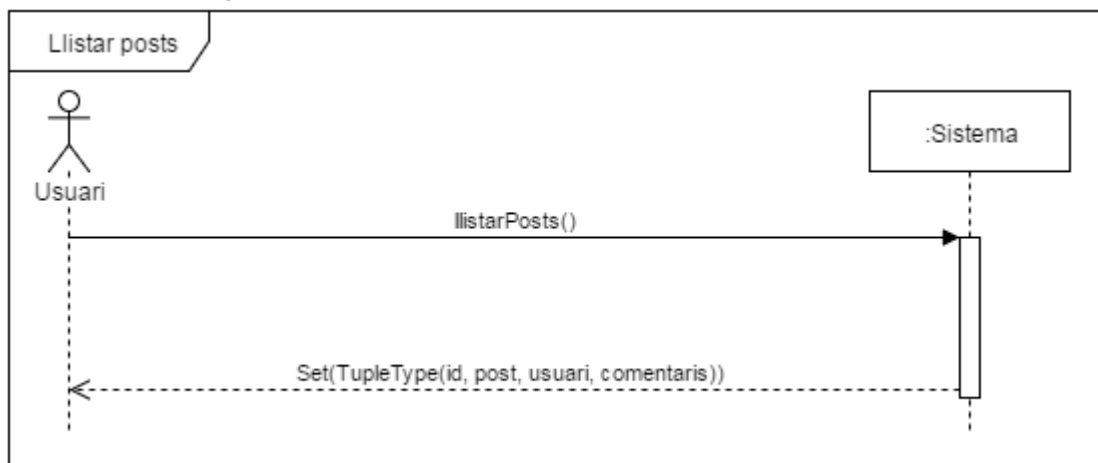
Context: Sistema::eliminarHotel(id: int)

Pre: L'usuari està logat.

Post: S'ha eliminat la instància d'Hotel *H* identificat amb *id* del sistema.

7.2.3 Mur

7.2.3.1 Llistar posts



Context: Sistema::llistarPosts() :Set(TupleType(*id*: int, *post*: string, *usuari*: usuari, *comentaris*: Set(TupleType(*comentari*: string, *usuari*: usuari))))

Pre: L'usuari està logat.

Post: L'usuari veu la llista de posts.

7.2.3.2 Crear post



Context: Sistema::crearPost(*post*: string, *usuari*: usuari)

Pre: L'usuari està logat, *post* no està buit, *usuari* és una instància de l'usuari logat que crea el post.

Post: S'ha creat una instància de Post *P* amb contingut *post*.

7.2.3.3 Editar post

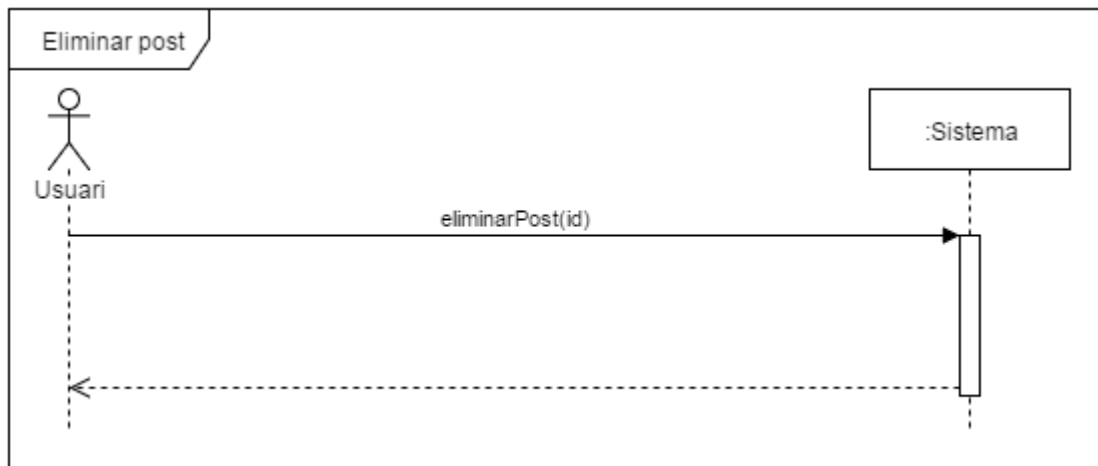


Context: Sistema::editarPost(*id*: int, *post*: string) :TupleType(*post*: string, *usuari*: usuari)

Pre: L'usuari està logat. Existeix un post amb identificador *id*.

Post: La instància de Post *P* identificada per *id* té com a contingut *post* i l'autor del post és *usuari*.

7.2.3.4 Eliminar post

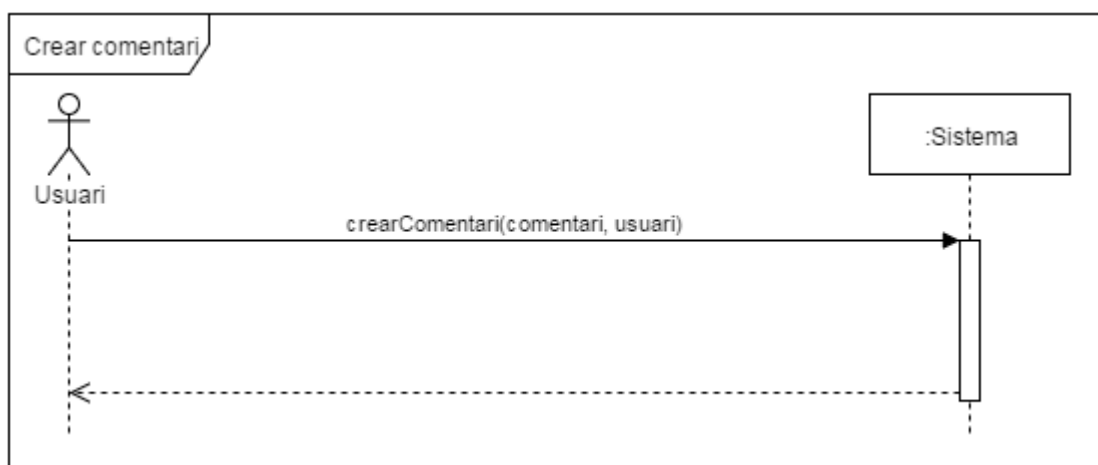


Context: Sistema::eliminarPost(*id*: int)

Pre: L'usuari està logat.

Post: S'ha eliminat la instància del Post *P* identificat amb *id* del sistema.

7.2.3.5 Crear comentari



Context: Sistema::crearComentari(*comentari*: string, *usuari*: usuari)

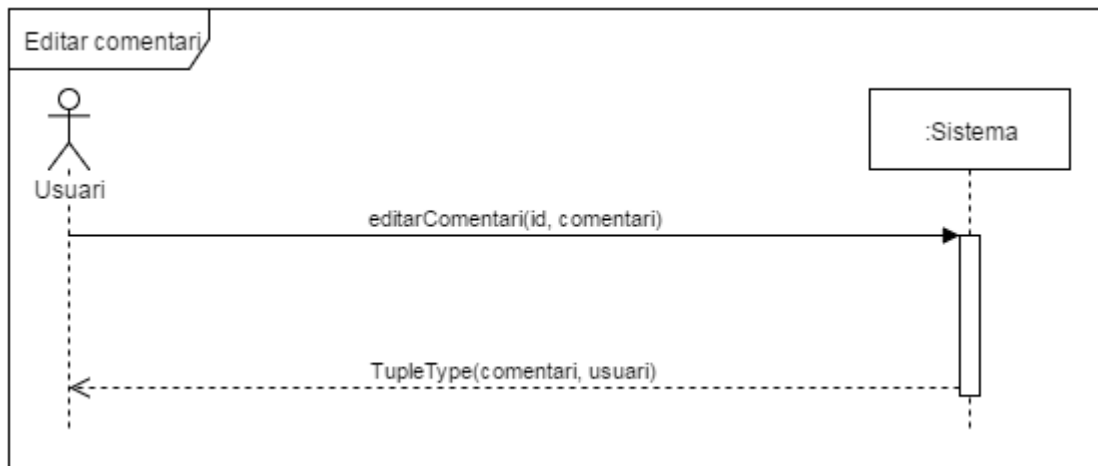
Pre: L'usuari està logat.

comentari no està buit.

usuari és la instància de l'usuari logat que crea el comentari.

Post: S'ha creat una instància de Comentari *C* amb contingut *comentari*.

7.2.3.6 Editar comentari



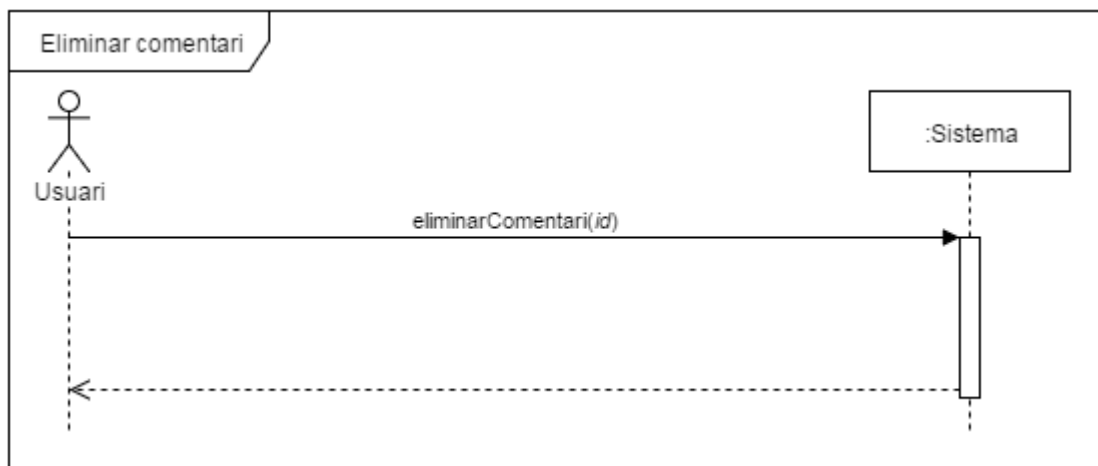
Context: Sistema::editarComentari(id: int, comentari: string) :TupleType(comentari: string, usuari: usuari)

Pre: L'usuari està logat.

Existeix un comentari amb identificador *id*.

Post: La instància de Comentari *C* identificada per *id* té com a contingut *comentari* i l'autor del comentari es *usuari*.

7.2.3.7 Eliminar comentari



Context: Sistema::eliminarComentari(id: int)

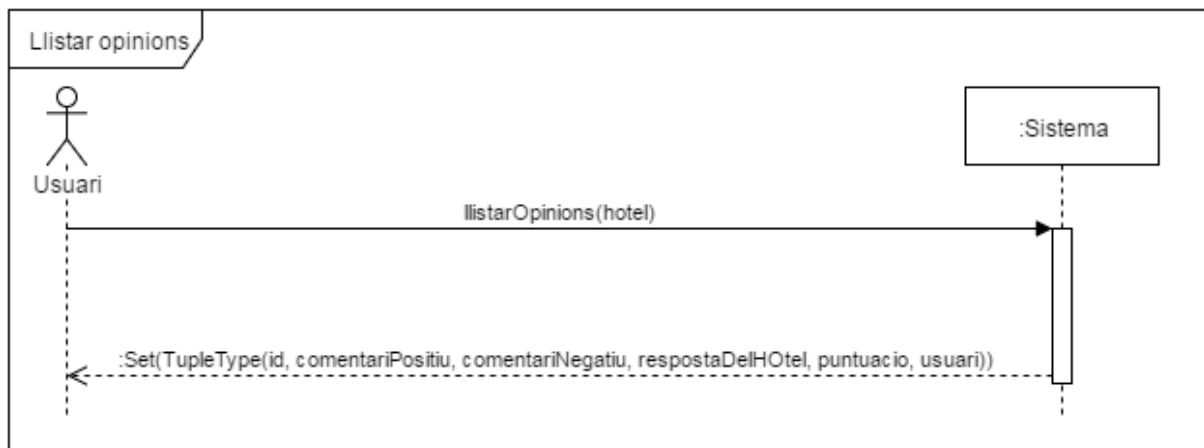
Pre: L'usuari està logat.

Existeix un comentari amb identificador *id*.

Post: S'ha eliminat la instància del Comentari *C* identificat amb *id* del sistema.

7.2.4 Opinions

7.2.4.1 Llistar opinions

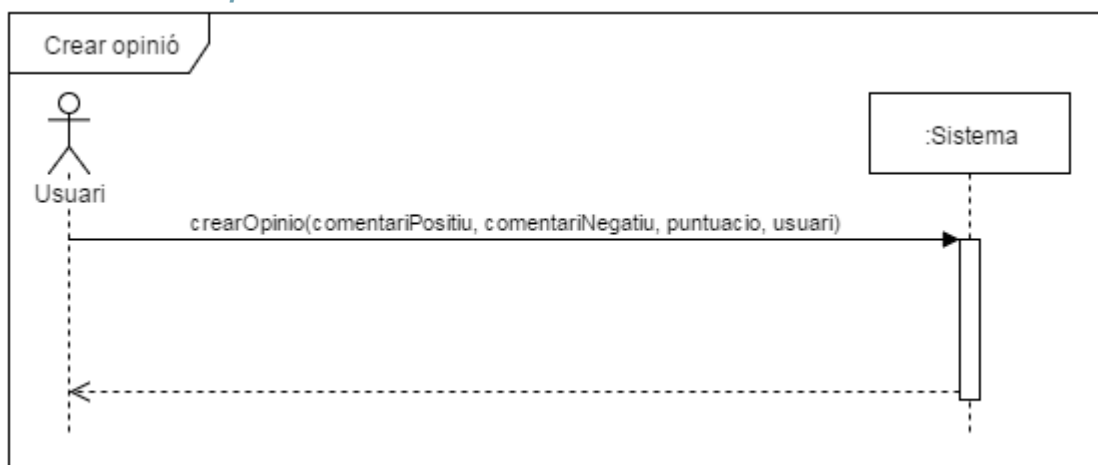


Context: Sistema::llistarOpinions(*hotel*: hotel) :Set(TupleType(*id*: int, *comentariPositiu*: string, *comentariNegatiu*: string, *respostaDelHotel*: string, *puntuació*: double, *usuari*: usuari))

Pre: L'usuari està logat.
Existeix una instància d'hotel *hotel*.

Post: L'usuari veu la llista d'opinions de l'hotel amb instància *hotel*.

7.2.4.2 Crear opinió

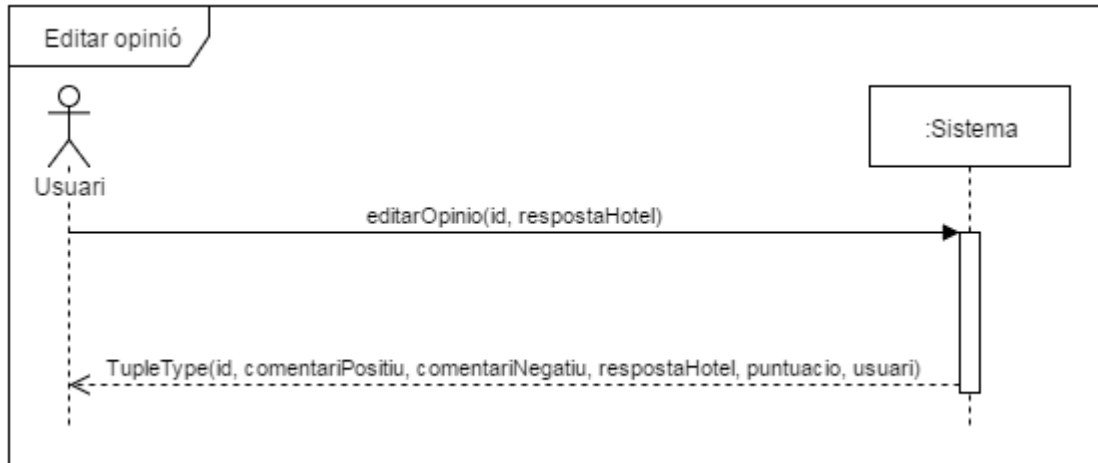


Context: Sistema::crearOpinio(*comentariPositiu*: string, *comentariNegatiu*: string, *puntuació*: double, *usuari*: usuari)

Pre: L'usuari està logat.
comentariPositiu, *comentariNegatiu* i *puntuació* no estan buits.
usuari és la instància de l'usuari logat que crea el comentari.

Post: S'ha creat una instància d'Opinió *O* amb un comentari positiu igual a *comentariPositiu*, un comentari negatiu igual a *comentariNegatiu* i amb una valoració de *puntuació*.

7.2.4.3 Editar opinió



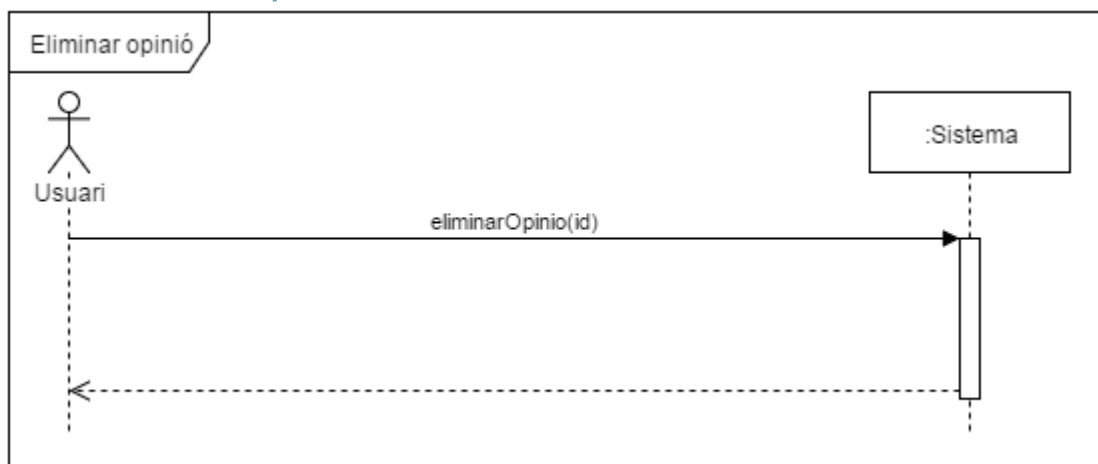
Context: Sistema::editarOpinio(id: int, *respostaHotel*: string) :TupleType(id: int, *comentariPositiu*: string, *comentariNegatiu*: string, *respostaHotel*: string, *puntuació*: double, *usuari*: usuari)

Pre: L'usuari està logat.

Existeix una opinió amb identificador *id*.

Post: La instància d'Opinió *O* identificada per *id* té com a resposta de l'hotel *respostaHotel*.

7.2.4.4 Eliminar opinió



Context: Sistema::eliminarOpinio(*id*: int)

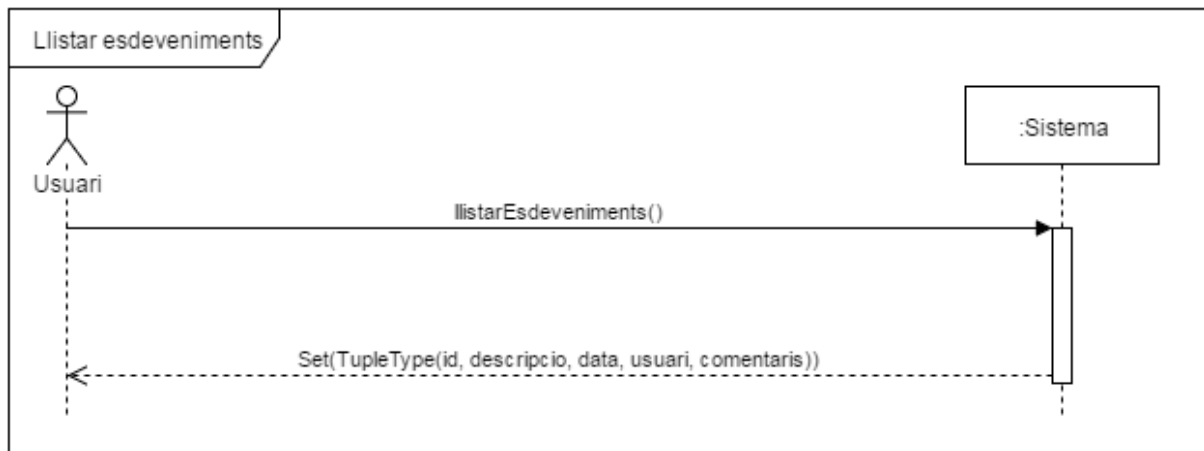
Pre: L'usuari està logat.

Existeix una opinió amb identificador *id*.

Post: S'ha eliminat la instància de la Opinió *O* identificat amb *id* del sistema.

7.2.5 Esdeveniments

7.2.5.1 Llistar esdeveniments



Context: Sistema::llistarEsdeveniments() :Set(TupleType(*id*: int, *descripció*: string, *data*: Date, *usuari*: usuari, *comentaris*: Set(*comentari*: comentari)))

Pre: L'usuari està logat.

Post: L'usuari veu la llista d'esdeveniments.

7.2.5.2 Crear esdeveniment



Context: Sistema::crearEsdeveniment(*descripció*: string, *data*: Date, *usuari*: usuari)

Pre: L'usuari està logat.

descripció i *data* no estan buits.

usuari identifica la instància de l'usuari que crea el comentari.

Post: S'ha creat una instància d'Esdeveniment *E* amb una descripció igual a *descripció* i una data igual a *data*.

7.2.5.3 Veure esdeveniment

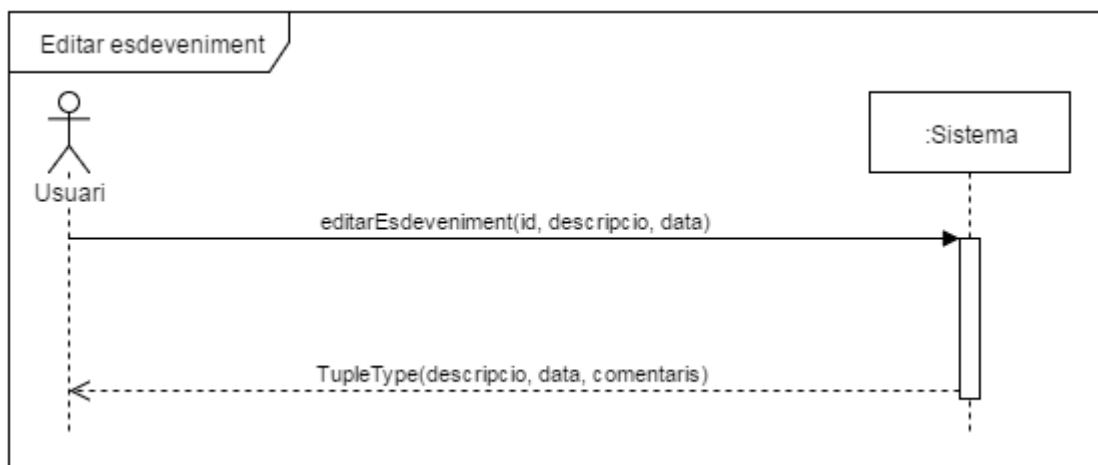


Context: Sistema::veureEsdeveniment(id: int) :TupleType(descripció: string, data: Date, comentaris: Set(comentari: comentari))

Pre: L'usuari està logat.

Post: L'usuari veu les dades de l'esdeveniment amb identificador *id*.

7.2.5.4 Editar esdeveniment



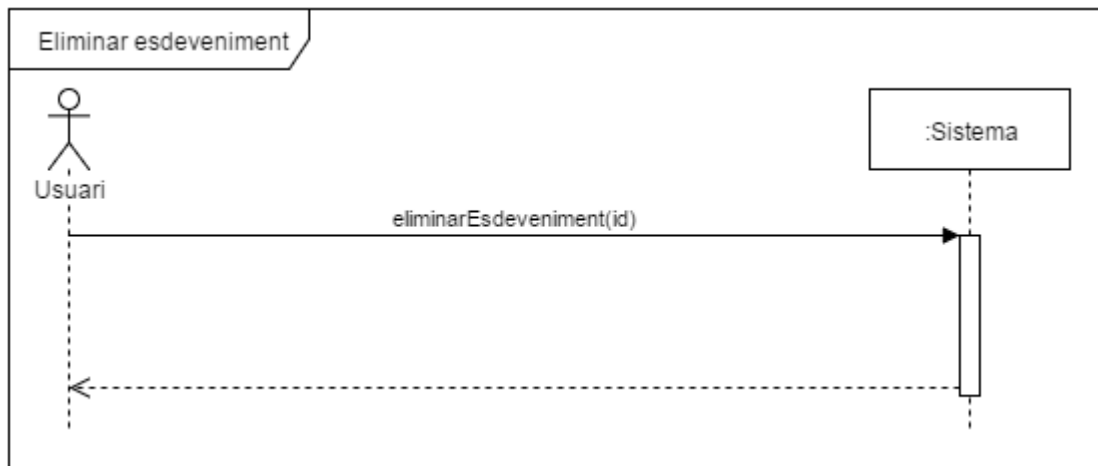
Context: Sistema::editarEsdeveniment(id: int, descripció: string, data: Date)
:TupleType(descripció: string, data: Date, comentaris: Set(comentari: comentari))

Pre: L'usuari està logat.

Existeix un esdeveniment amb identificador *id*.

Post: La instància d'Esdeveniment *E* identificada per *id* té com a descripció *descripció* i com a data *data*.

7.2.5.5 Eliminar esdeveniment



Context: Sistema::eliminarEsdeveniment(*id*: int)

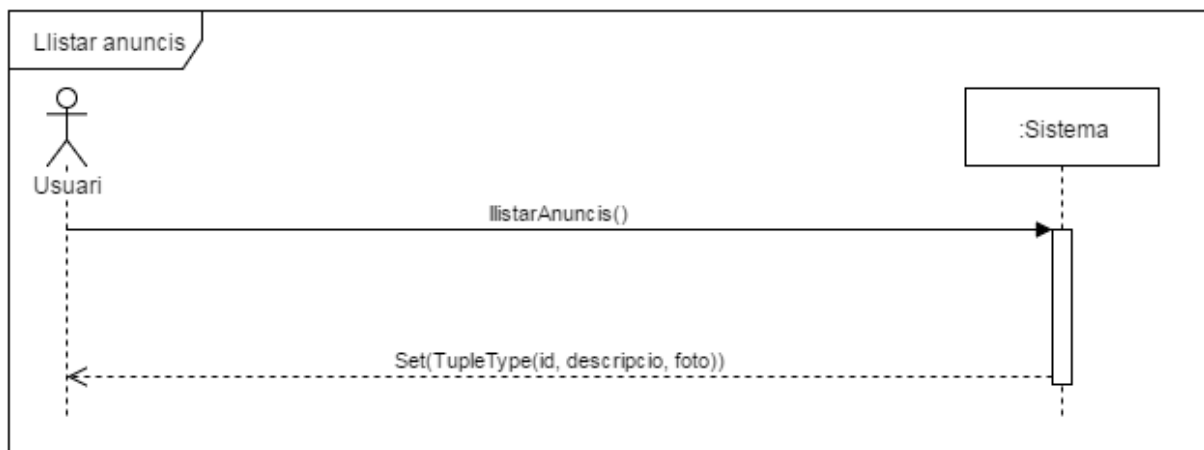
Pre: L'usuari està logat.

Existeix un esdeveniment identificat amb *id*.

Post: S'ha eliminat la instància de l'Esdeveniment *E* identificat amb *id* del sistema.

7.2.6 Anuncis

7.2.6.1 Llistar anuncis



Context: Sistema::llistarAnuncis() :Set(TupleType(*id*: int, *descripció*: string, *foto*: image))

Pre: L'usuari està logat.

Post: L'usuari veu la llista d'anuncis.

7.2.6.2 Crear anunci

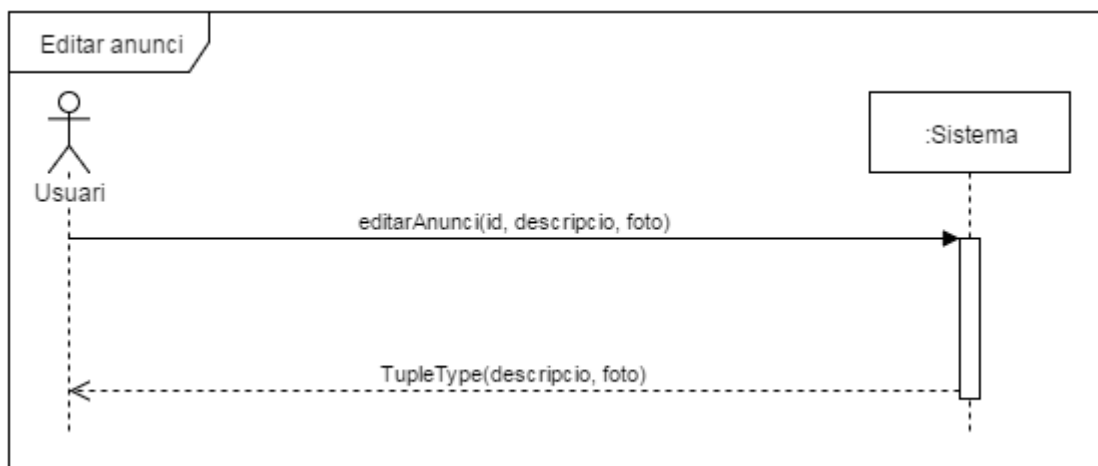


Context: Sistema::crearAnunci(*descripció*: string, *foto*: image)

Pre: *descripció* i *foto* no estan buits.

Post: S'ha creat una instància d'Anunci A amb una descripció igual a *descripció* i una imatge igual a *foto*.

7.2.6.3 Editar anunci



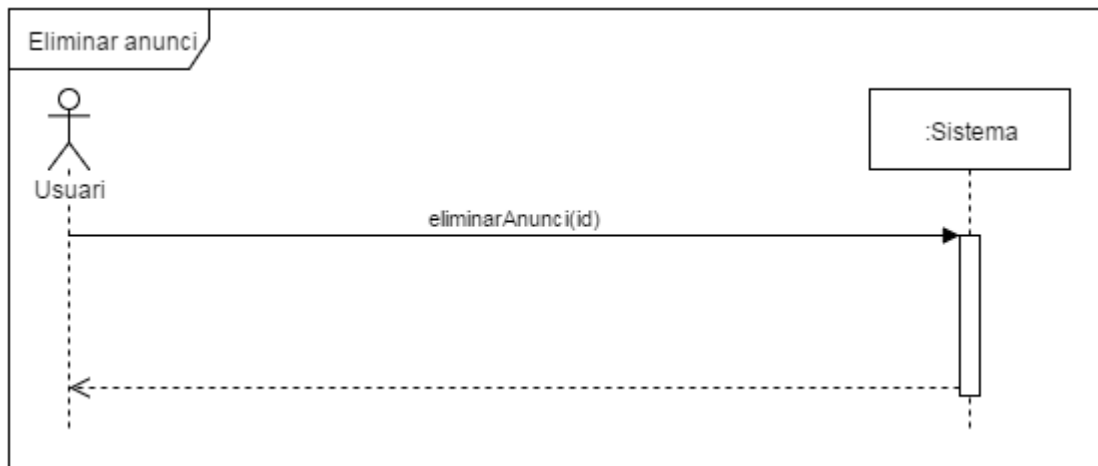
Context: Sistema::editarAnunci(*id*: int, *descripció*: string, *foto*: image)
:TupleType(*descripció*: string, *foto*: image)

Pre: L'usuari està logat.

Existeix un anunci amb identificador *id*.

Post: La instància d'Anunci A identificada per *id* té com a descripció *descripció* i com a imatge *foto*.

7.2.6.4 Eliminar anunci



Context: Sistema::eliminarAnunci(*id*: int)

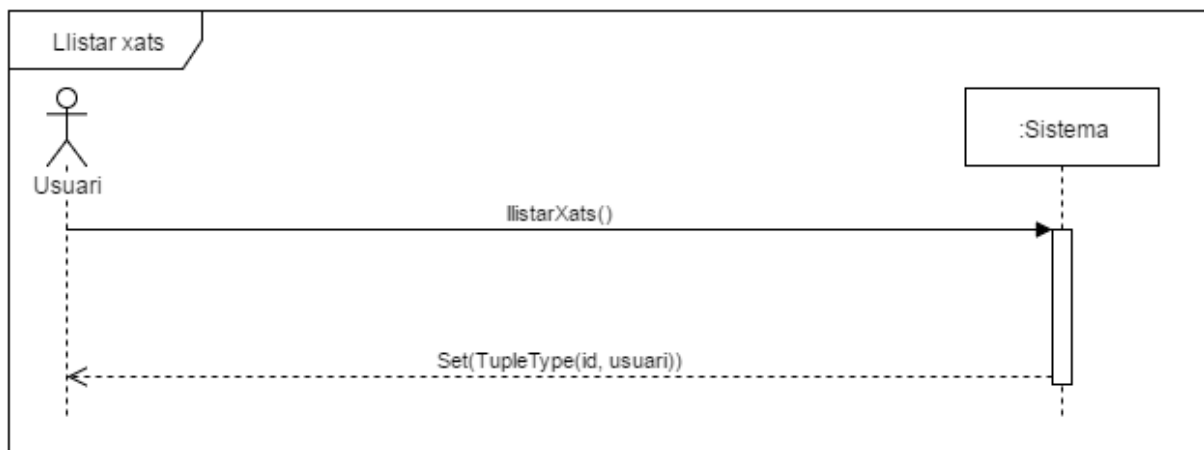
Pre: L'usuari està logat.

Existeix un anunci amb identificador *id*.

Post: S'ha eliminat la instància de l'Anunci A identificat amb *id* del sistema.

7.2.7 Missatges

7.2.7.1 Llistar xats



Context: Sistema::llistarXats() :Set(TupleType(*id*: int, *usuari*: usuari))

Pre: L'usuari està logat.

Post: L'usuari veu la llista de xats amb altres usuaris.

7.2.7.2 Veure xat

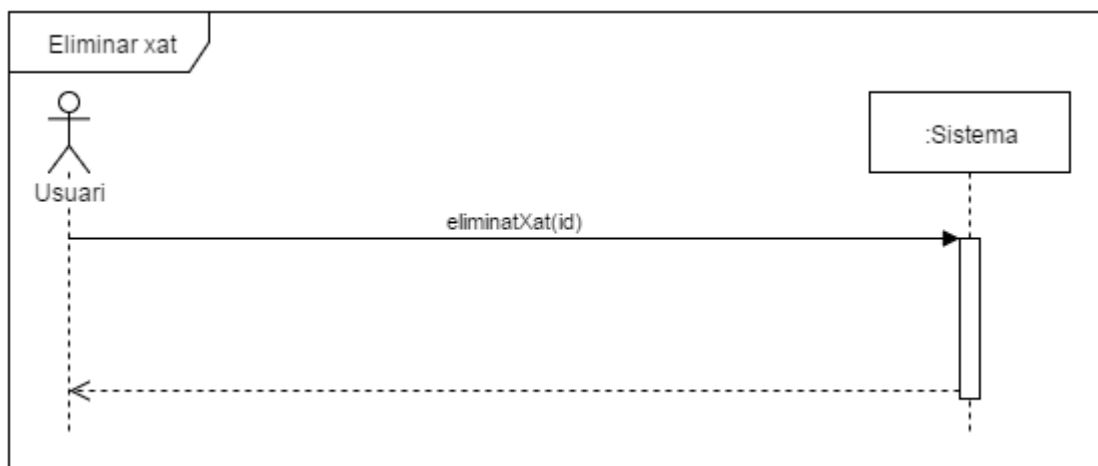


Context: Sistema::veureXat() :Set(TupleType(*usuari*: usuari, *missatges*: Set(TupleType(*missatge*: string, *data*: Date))))

Pre: L'usuari està logat.

Post: L'usuari veu la llista de missatges *missatges* intercanviats amb l'usuari *usuari* ordenats per la data *data*.

7.2.7.3 Eliminar xat



Context: Sistema::eliminatXat(*id*: int)

Pre: L'usuari està logat.

Post: La instància de Xat X identificada amb *id* es marca com a eliminat.

7.2.7.4 Enviar missatge



Context: Sistema::enviarMissatge(*emissor*: usuari, *receptor*: usuari, *missatge*: string)

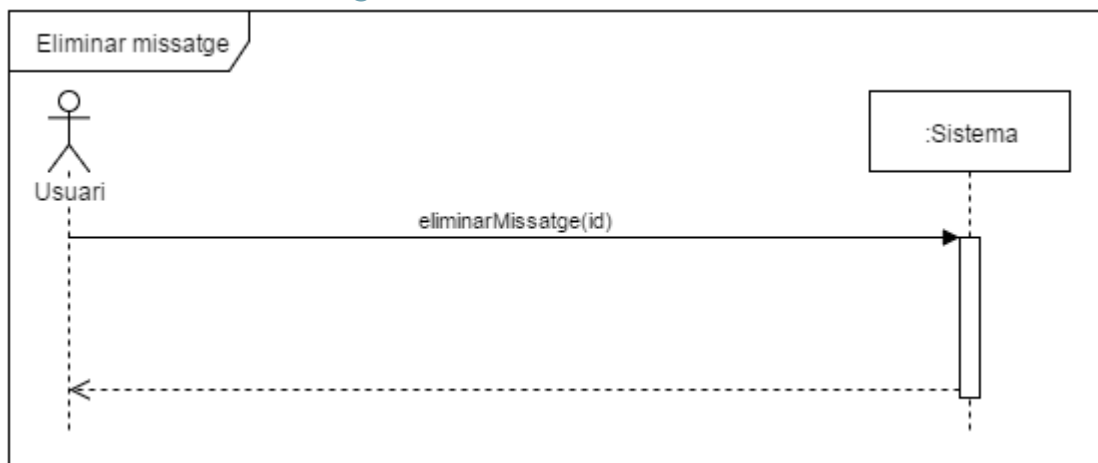
Pre: L'usuari està logat.

emissor és la instància de l'usuari que està logat i envia el missatge.

Existeix una instància d'usuari *receptor* diferent a *emissor*.

Post: Es crea una instància de Missatge M amb usuari emissor *emissor* i un usuari receptor *receptor* amb un contingut *missatge*.

7.2.7.5 Eliminar missatge



Context: Sistema::eliminarMissatge(*id*: int)

Pre: L'usuari està logat.

Existeix un missatge amb identificador *id*.

Post: La instància de Missatge M identificada amb *id* es marca com a eliminada.

8 Disseny

8.1 Arquitectura

A continuació llistaré els requisits no funcionals que m'han fet prendre alguna decisió sobre l'arquitectura que tindrà el projecte.

RNF-02: El sistema serà accessible des dels navegadors més utilitzats. Degut a aquesta restricció l'aplicació ha de ser tipus web i tenir la **capa de presentació** el més desacoblada possible, ja que un error a un navegador es pugui corregir modificant la capa de presentació.

RNF-04: El sistema ha de ser capaç de ser testejat a tots els nivells: Visual, usabilitat, accessibilitat, codi i rendiment. Per poder testejar de manera separada hem de tenir el codi **separat en capes**, de no ser així no sabríem detectar l'origen de l'error.

Arribats a aquest punt he decidit fer l'aplicació en tres capes: Presentació, domini i dades. D'aquesta manera separem les funcionalitats de l'aplicació en nivells i cada nivell s'encarrega de fer les seves funcionalitat i si no les ha de fer les delega a un altre nivell.

Presentació

Aquesta capa és la responsable de la interacció amb l'usuari; sap la forma en que ha de presentar les dades a l'usuari que rep de la capa de domini.

Domini

Aquesta capa s'encarrega de la implementació de les funcionalitats del sistema; és transparent i inaccessible per l'usuari. Rep les peticions de la capa de presentació, valida les dades d'entrada, efectua accions i realitza consultes a la capa de dades. Finalment envia el resultat a la capa de presentació.

Dades

És on resideixen les dades i encarregada d'accedir a elles. Està formada per un gestor de base de dades que realitza tot l'emmagatzematge de dades. Rep peticions de la capa de domini i retorna les dades pertinents.

8.1.1 Tecnologia utilitzada

A continuació llistaré els requisits no funcionals que m'han fet prendre alguna decisió sobre la tecnologia que tindrà el projecte. També tindrà una gran influència en l'arquitectura de l'aplicació.

RNF-12: El sistema ha de ser flexible a futurs canvis i apte per a la incorporació de noves funcionalitats. Com més desacoblat tinguem el codi més fàcil serà realitzar un canvi ja que serà més fàcil testejar-lo. Per això proposo una programació seguint el patró de **Model – Vista – Controlador**. Les vistes formen la capa de presentació i el model i els controladors formen a capa de domini.

RNF-09: El sistema ha d'estar dissenyat per a que consumeixi el mínim possible de dades, recursos i temps. Per poder reduir el nombre de dades consumides introduïrem **sistemes de cache**, un per peticions HTTP i un altre per emmagatzemar dades de la base de dades i evitar lectures innecessàries. Això ens aportarà velocitat i per tant una reducció de temps i dades consumides.

RNF-08: El sistema haurà de ser capaç d'atendre sol·licituds segons la demanda. Per poder assolir la demanda de peticions haurem de tenir un sistema escalable. Per tal d'escalar de la manera més eficient, separarem el **sistema amb capes**. Tindrem cada una de les capes en servidors separats. Així escalarà el conjunt de servidors necessari. Per exemple, si les peticions són a la base de dades, escalaran els servidors de bases de dades, i no hi haurà necessitat d'escalar el servidors de presentació o de domini. El mateix pot passar en les altres capes. El millor hosting per escalar aplicacions és **Amazon Web Services**.

RNF-01: El sistema no podrà estar inaccessible mentre s'estigui actualitzant. Per poder aconseguir això, farem servir un software de desplegament com **Ansible** (<https://www.ansible.com/>). Així podrem desplegar el codi al servidor amb seguretat i certesa. Un cop estigui desplegat es podrà fer el canvi del codi antic al nou.

RNF-05: El sistema ha de tenir un procediment documentat de desplegament de l'aplicació tant per a servidors de desenvolupament com de producció. Per assolir aquest requeriment farem servir **Ansible**, ja que la manera que té per emmagatzemar els procediments de desplegament s'anomenen receptes i són fàcilment llegibles i autodocumentables.

RNF-15: El sistema ha de disposar d'un procediment en cas de desastre per a tornar a posar-lo en funcionament el més aviat possible. Si escollim **Amazon Web Services** com a proveïdor de servidors no tindrem problemes de discs ja que tot va per instàncies virtuals i si un servidor físic cau t'assignen un altre automàticament. En el pitjors dels casos el que podria passar és que tota una zona d'Amazon estés fora de servei. Per solucionar això, utilitzariem **Ansible** per fer el desplegament a Amazon indicant una altra zona de disponibilitat.

RNF-03: El sistema haurà de tenir capacitat per emmagatzemar dades segons la demanda. Per assolir aquest requisit haurem de separar la capa de dades de la resta per si ha d'escalar, ja sigui verticalment o horitzontalment. Degut a això he escollit **Amazon Web Services** (<https://aws.amazon.com>) per a hostejar l'aplicació.

RNF-06: El sistema estarà disponible 24 hores al dia els 365 dies de l'any. Necessitarem que l'aplicació estigui hostejada en uns servidors d'alta disponibilitat. Avui en dia la majoria de hostings són fiables, però el rei del sector és **Amazon Web Services**.

RNF-10: El sistema realitzarà còpies de seguretat constantment. Si tenim el **model de dades** separat de la resta del sistema i ho tenim allotjat a **Amazon Web Services** en una instància tipus RDS ja tenim cobert aquest requisit, ja que RDS realitza còpies de seguretat automàticament.

8.1.1.1 Django

La primera tecnologia escollida és Django, el qual és framework web, escrit amb Python, i amb ell s'ha realitzat el nucli de l'aplicació (<https://www.djangoproject.com/>).

Un dels motius pel qual he escollit aquesta tecnologia és perquè és Open Source i això implica que té una gran comunitat al darrera. Gràcies a això, Django disposa d'una documentació actualitzada constantment amb molts exemples i tutorials. Podem trobar més de 100.000 preguntes i respostes a StackOverflow (<http://stackoverflow.com/>) sobre Django.

Amb una comunitat tan ampla és més ràpid detectar errors i que els usuaris els puguin corregir. També poden modificar el codi per adaptar-lo a les seves necessitats. Ja que el projecte és públic a GitHub es van creant peticions d'actualitzacions i així s'aconsegueix un codi robust, reusable i portable.

Una alternativa a modificar el codi font és la creació de llibreries o extensions del propi framework. Alhora són Open Source i mantingudes per la comunitat. Existeixen una gran varietat d'extensions per a Django, i una d'elles és Django Rest Framework (<http://www.django-rest-framework.org/>) que ens permet construir una API REST amb els recursos que ofereix Django.

Un dels altres motius pels quals he escollit Django és que segueix el patró Model – Vista – Controlador. En aquest cas els controladors els anomena Views i a les vistes Templates.

Les vistes de Django (Templates) són un mòdul independent del framework fàcilment reemplaçable, que per defecte utilitza Django Templates. Parteixen de la base de l'HTML i afegeixen una anotació d'etiquetes que es reemplaça abans de ser renderitzat l'HTML pel servidor web.

Django disposa d'un ORM (Object-Relational Mapping) que ens permet mapejar taules de la base de dades (en el nostre cas utilitzarem PostgreSQL) amb objectes de la nostra aplicació. Addicionalment incorpora un mòdul que ens facilita la creació, edició i eliminació de taules i columnes de la base de dades únicament modificant els objectes de l'aplicació. Gràcies a això, podem despreocupar-nos de la gestió de la base de dades.

L'organització del codi en comptes d'estar estructurada per models, vistes i controladors està organitzada en aplicacions. Cada aplicació pot ser independent a les altres, es pot abstrure i ser reutilitzada per a altres projectes.

Finalment ens proporciona totes les llibreries necessàries per poder realitzar tests unitaris del codi.

Alguns exemples d'aplicacions que utilitzen aquest framework són Instagram, Pinterest o Disqus

8.1.1.2 Python

Python és un llenguatge de programació molt senzill i net. Moltes de les llibreries que pot utilitzar Django no són exclusives al framework sino que al estar fetes amb Python poden ser utilitzades per altres *frameworks* o programes i això fa que tinguin una comunitat al darrera més gran.

8.1.1.3 HTML + CSS

La part de les vistes les farem amb Django Templates, que utilitza com a base HTML.

L'HTML és el llenguatge utilitzat per a realitzar pàgines web estàtiques. Ens permet estructurar les dades utilitzant etiquetes. En aquestes etiquetes hi podem afegir atributs per a modificar el seu comportament i crear elements com imatges, formularis, botons, taules... També hi podem afegir identificadors i classes que seran utilitzades pel CSS.

El CSS ens permet donar estil a l'HTML. A partir dels identificadors i les classes pot donar forma, color, mida... a les etiquetes seleccionades.

8.1.1.4 jQuery

És tracta d'una llibreria de JavaScript (<https://jquery.com/>) que simplifica les seves funcionalitats. En el nostre cas l'utilitzarem per a crear animacions visuals, mostrar algun mapa de Google Maps i sobretot per a realitzar crides AJAX al servidor.

8.1.1.5 Bootstrap

Ja que uns dels nostres requeriments és que l'aplicació es pugui utilitzar a qualsevol dispositiu he escollit utilitzar Bootstrap (<http://getbootstrap.com/>), una llibreria frontend per a poder fer-lo adaptatiu.

A més a més, ens aporta elements HTML ja dissenyats per a utilitzar-los a les nostres vistes sense preocupar-nos de quin dispositiu el renderietza.

8.1.1.6 PostgreSQL

Degut a que el projecte requereix guardar moltes relacions entre objectes, sobretot amb usuaris, he optat per una base de dades relacional en comptes d'una documental.

Entre les bases de dades relacionals de programari lliure que hi ha, Django ens ofereix escollir entre MySQL o PostgreSQL.

Em decanto per PostgreSQL pel fet de ja haver treballat abans amb MySQL i no estic gens satisfet amb els seus resultats..

8.1.1.7 Redis

Redis (<http://redis.io/>) és un motor de base de dades de tipus clau valor que emmagatzema les dades en memòria o disc. En el nostre projecte l'utilitzarem com a caché de l'aplicació per evitar realitzar peticions a la base de dades que serà el coll d'ampolla.

Una possible millora de l'aplicació seria utilitzar Redis com a base de dades documental en algunes part de la web, com per exemple el llistat de posts o el llistat d'usuaris.

8.1.1.8 RabbitMQ

És un software de negociació de missatges (<https://www.rabbitmq.com/>). A la nostra aplicació ho utilitzarem per gestionar tasques més pesades que volem fer en segon pla amb l'ajuda de la llibreria Celery (<http://www.celeryproject.org/>). Celery és un gestor de cues que també ens ajuda a programar tasques asíncrones.

Per altra banda utilitzarem Rabbit per a l'intercanvi de missatges entre usuaris.

8.1.1.9 NginX

NginX (<https://www.nginx.com/>) és un servidor web, com ho podia ser Apache, però en el nostre cas ho utilitzaré com a proxy invers o load balancer. S'encarregarà de rebre totes les peticions d'Internet en HTTPS, les descriptarà i l'enviarà en el següent nivell de la pila. També serà l'encarregat de servir els elements estàtics com imatges o fitxers.

8.1.1.10 Varnish

Varnish (<https://www.varnish-cache.org/>) és un accelerador d'aplicacions web o caché de peticions HTTP. Es situa al davant del servidor web i guarda una còpia del recurs sol·licitat.

8.1.1.11 uWSGI

De la llista de servidors webs que podem trobar he escollit uWSGI (<https://uwsgi-docs.readthedocs.io/en/latest/>) ja que es el que és recomana per a projectes Django, és lleuger i ens permet gestionar varis processos.

8.1.1.12 Ubuntu

Com a sistema operatiu, tant com a la màquina de desenvolupament com al servidor de producció utilitzarem Ubuntu (<http://www.ubuntu.com/server>). És un sistema operatiu basat en Linux el qual suporta tots els paquets prèviament nombrats.

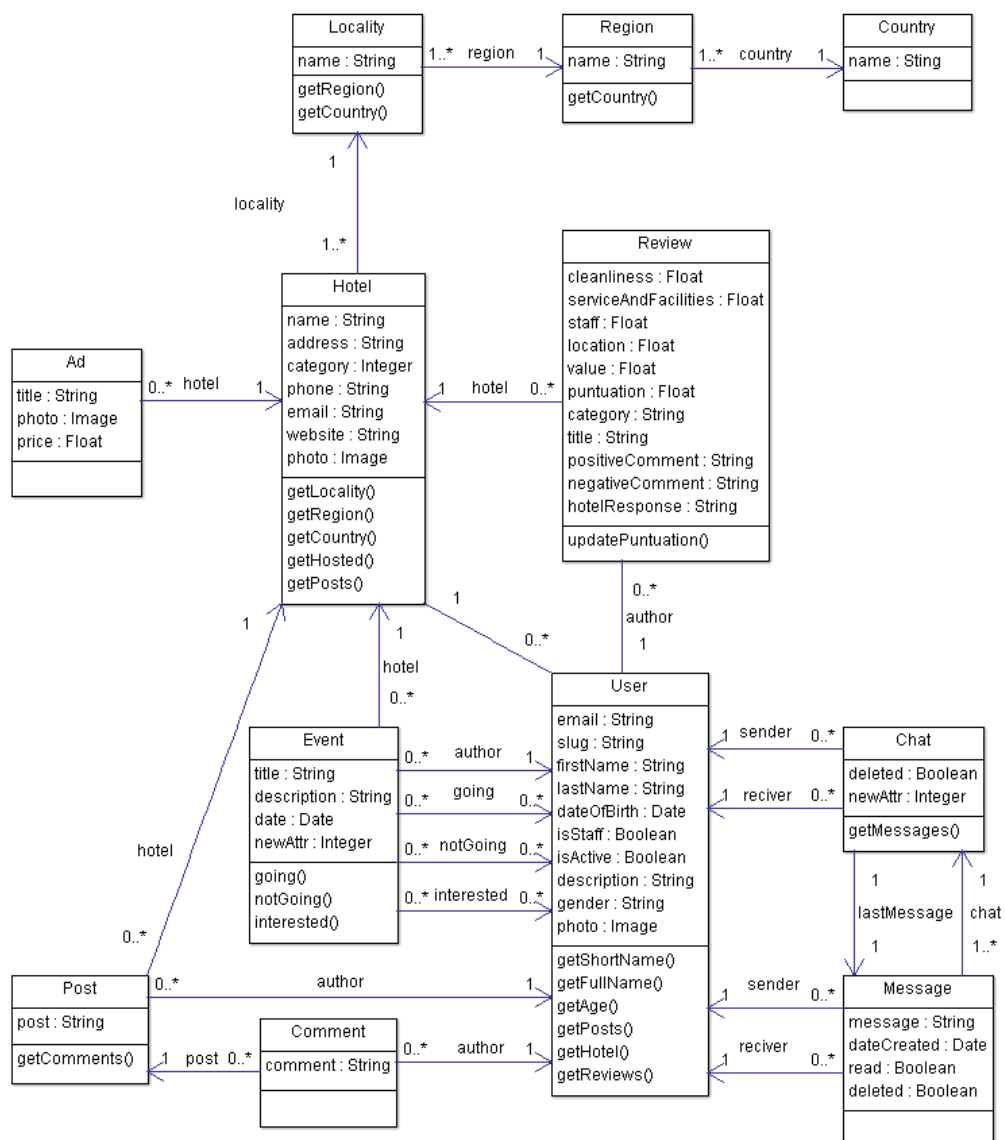
8.1.1.13 Ansible

A l'hora de configurar el servidor i desplegar el codi farem servir Ansible (<https://www.ansible.com/>). Amb la simple configuració d'uns fitxers amb YML podem desplegar el codi en un servidor remot que prèviament hem configurat amb el mateix Ansible.

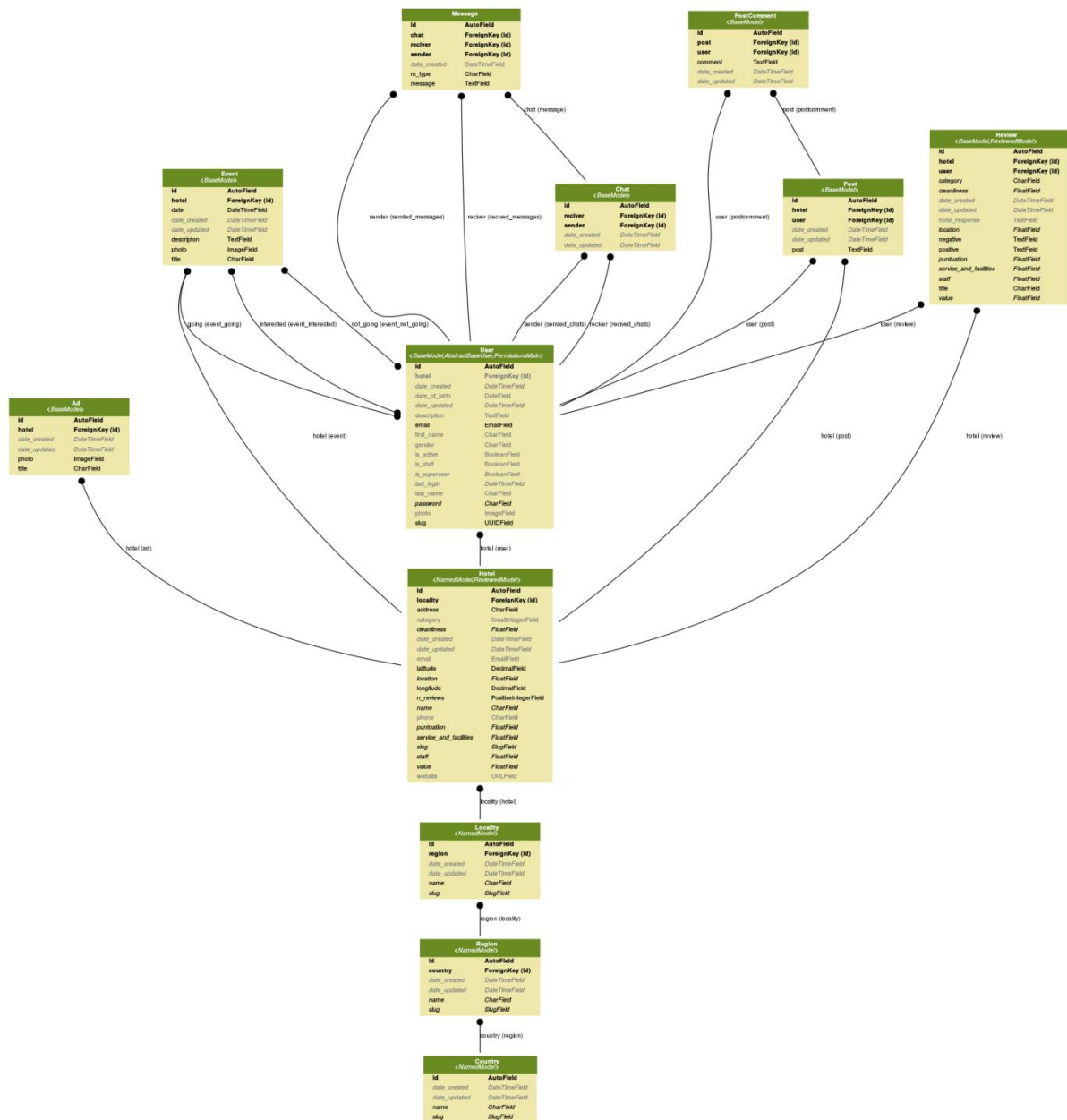
La configuració i connexions finals dels servidor quedarien així:



8.2 Diagrama de classes de disseny



8.3 Model de dades



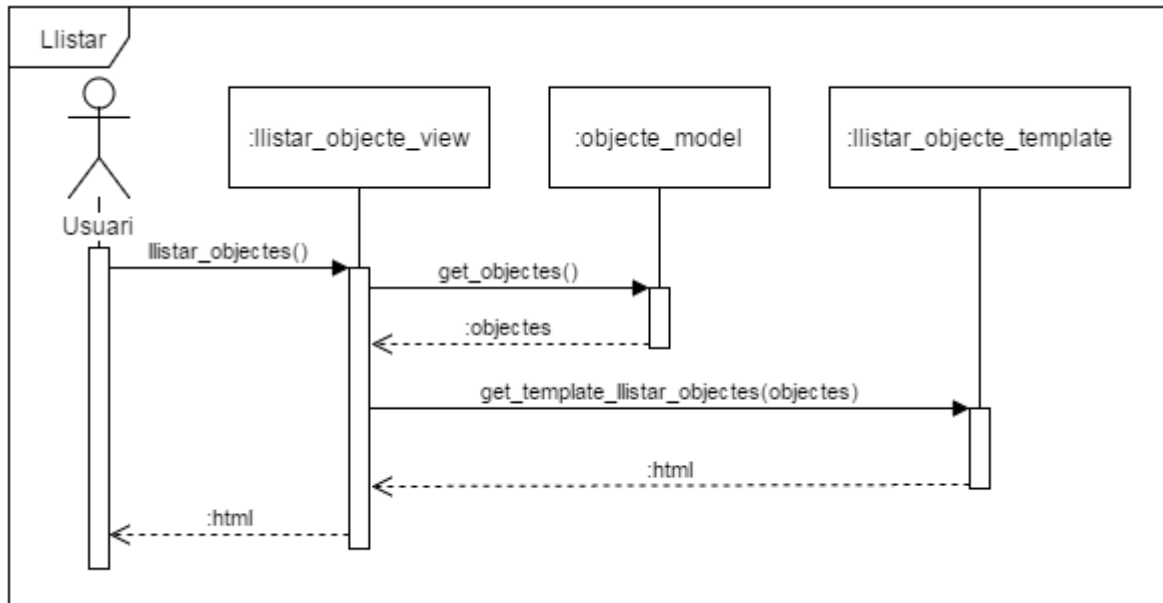
8.4 Diagrames de seqüència

Per evitar redundància en aquesta part he intentat simplificar els diagrames i mostrar els que són comuns d'una manera genèrica.

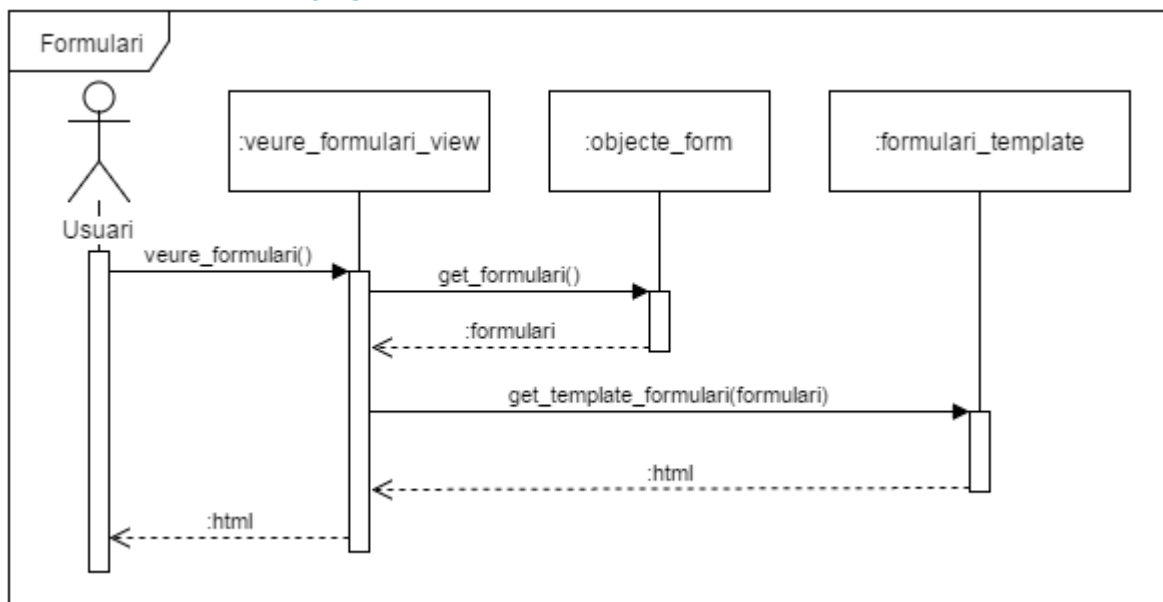
Quan mencionem Objecte ens referim a Hotel, Usuari, Post, Comentari, Opinió, Esdeveniment, Anuncio, Xat o Missatge.

A més a més, inclourem alguns diagrames més específics con els de Login i Logout.

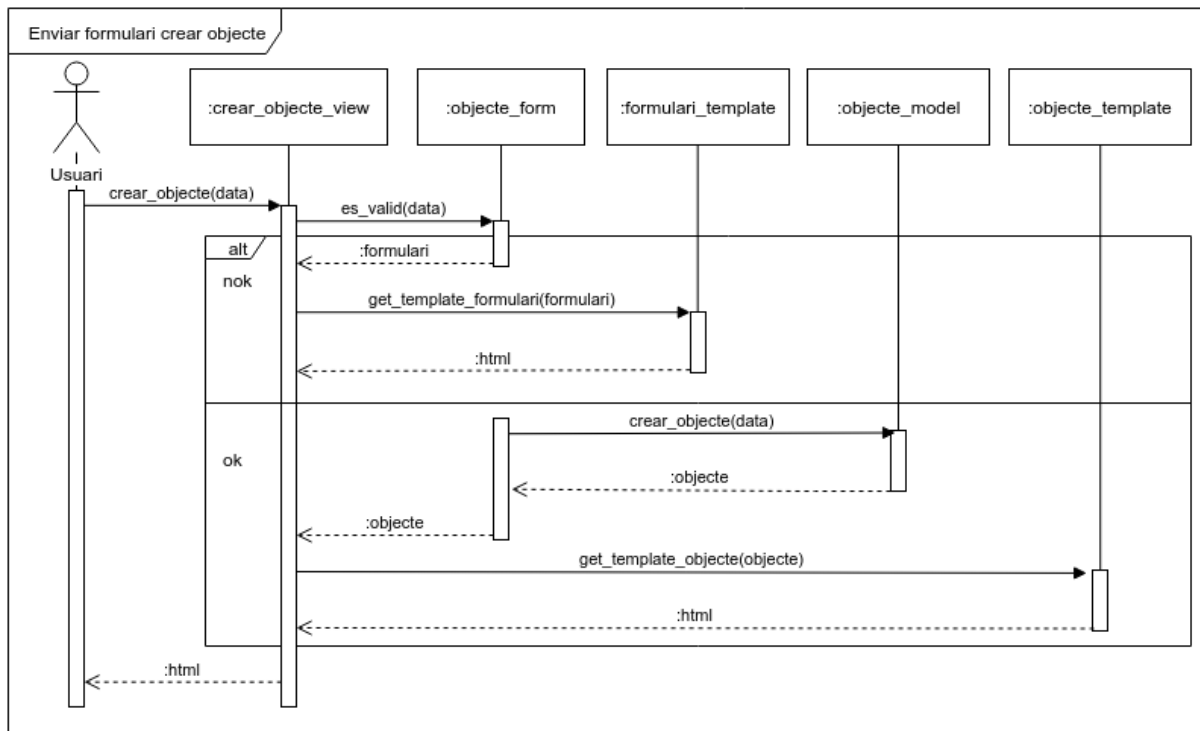
8.4.1 Accedir a un llistat d'Objectes



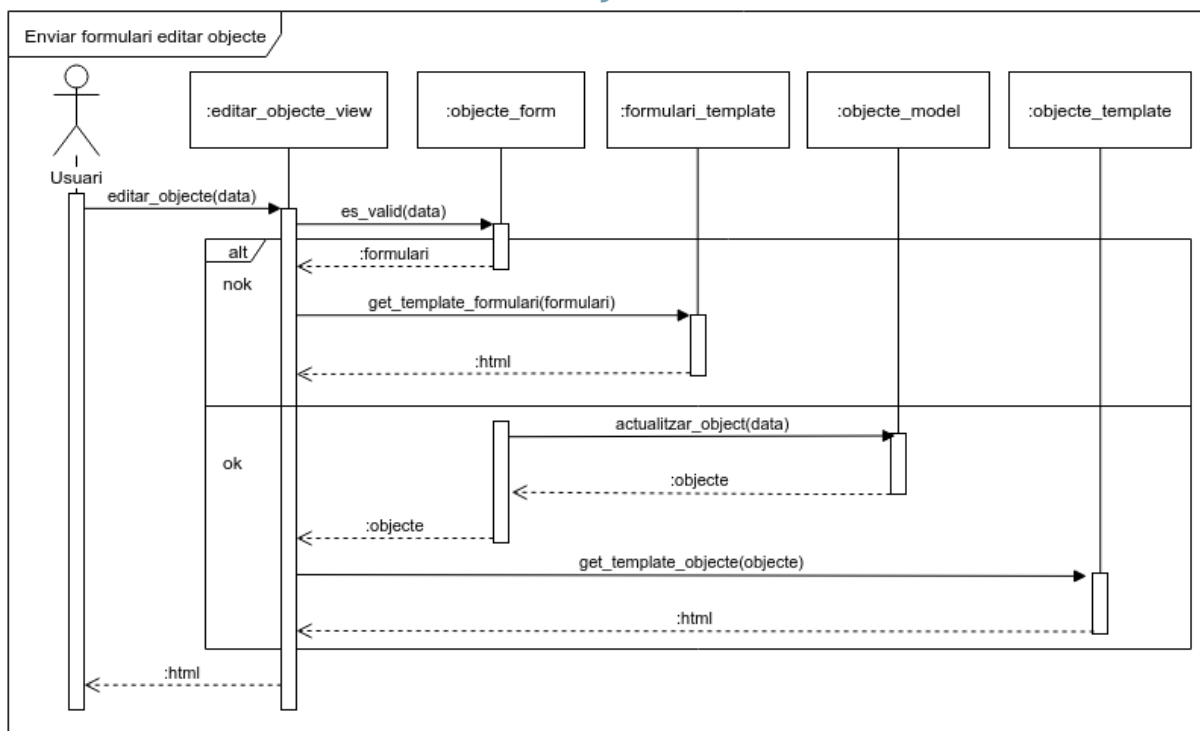
8.4.2 Accedir a una pàgina amb formulari



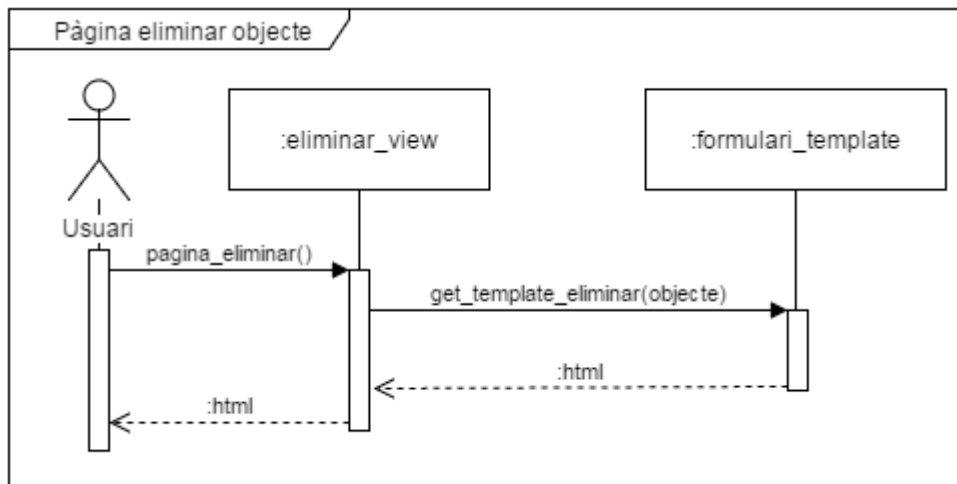
8.4.3 Enviar un formulari de creació d'Objecte



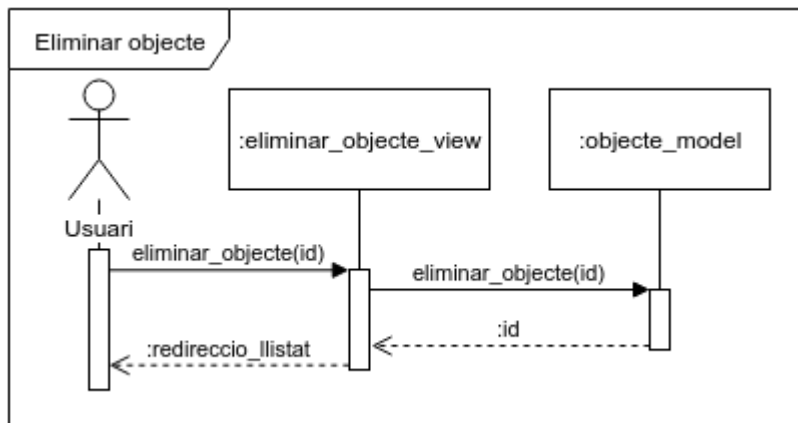
8.4.4 Enviar un formulari d'edició d'Objecte



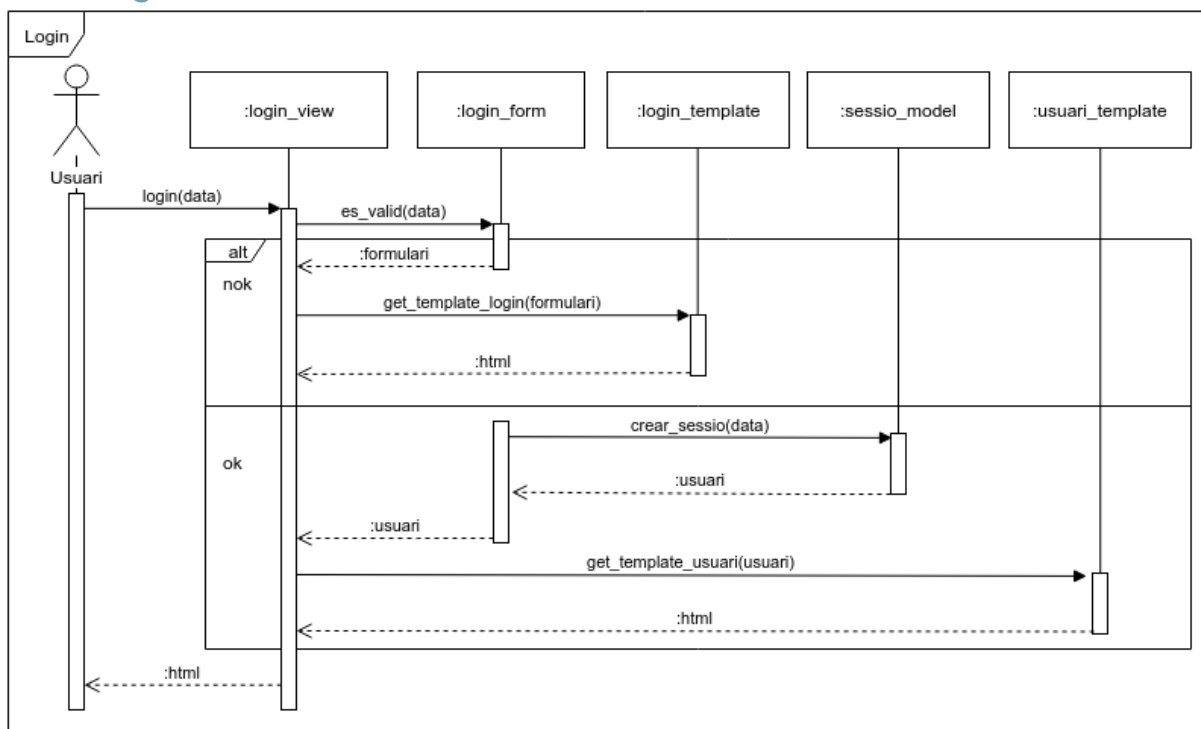
8.4.5 Accedir a la pàgina d'eliminació d'Objecte



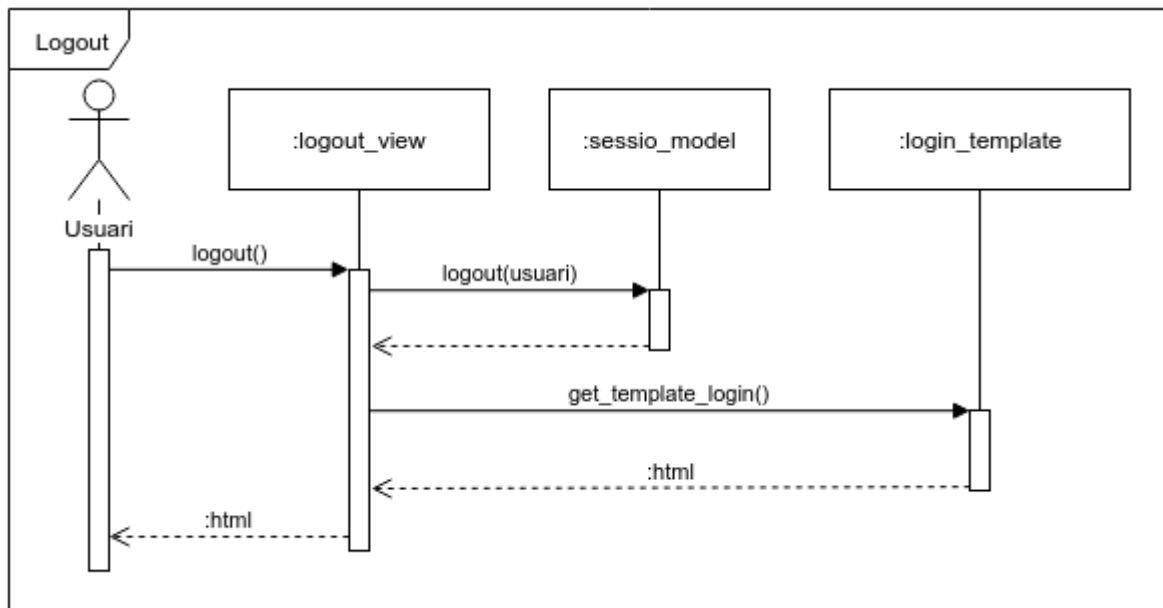
8.4.6 Enviar petició d'eliminació d'Objecte



8.4.7 Login



8.4.8 Logout



9 Implementació i Proves

9.1 Instal·lació d'un projecte Django

En aquest apartat explicaré com s'instal·la *Django* i ensenyaré com funciona tot el flux per a la creació d'un cas d'ús.

És recomanable seguir el tutorial que trobarem a la pàgina oficial de *Django* (<https://docs.djangoproject.com/en/1.9/intro/install/>).

Suposarem que treballem en una màquina *Linux*, en el meu cas amb Ubuntu, però és igualment vàlid per a un sistema *MacOSX* o *Windows*.

Per començar instal·lem *Django* amb la comanda:

```
pip install Django
```

Suposarem que tenim instal·lada una versió de *Python* superior a la 2.7. Ens descarregarà i instal·larà *Django* a la nostra màquina. És recomanable utilitzar *Virtualenv* (<https://virtualenv.pypa.io/en/latest/>) per a protegir el nostre entorn de la resta del sistema operatiu.

Tot seguit podem executar la següent comanda per a la creació de l'estructura bàsica de directoris per a poder començar un projecte:

```
django-admin startproject hotel_social_network
```

L'estructura del projecte seria la següent:

```
src/  
    manage.py  
    hotel_social_network/  
        settings.py  
        urls.py  
        templates/
```

El fitxer `manage.py` ens permet executar scripts del projecte per a automatitzar tasques dins del nostre entorn de feina. Exemples que utilitzarem són:

- `startapp`: Ens crea una aplicació dins del nostre arbre de carpetes del projecte.
- `createsuperuser`: Crea un usuari amb permisos de superadministrador a la base de dades del projecte.
- `shell`: Ens mostra una consola de comandes *Python* utilitzant l'entorn actual de treball.
- `graph_models`: Exporta un gràfic amb les taules actuals de la base de dades.
- `check`: Comprova que tots els fitxers compilin i no generin cap error.
- `runserver`: Executa un servidor web de desenvolupament.

- `makemigrations`: Genera els fitxers amb els canvis que s'han d'aplicar a la base de dades amb els canvis dels models.
- `migrate`: Aplica els canvis generats per `makemigrations`.

El fitxer de `settings.py` conté els paràmetres de configuració del nostre projecte com per exemple bases de dades, llistat d'aplicacions, zona horària, idioma del projecte...

El fitxer `urls.py` conté els punts d'entrada de la nostra aplicació. Són expressions regulars que filtren les peticions HTTP i decideix quin controlador s'ha d'executar tot seguit.

Finalment el directori `templates/` conté els directoris amb fitxers HTML de la nostra aplicació, és a dir, les vistes.

9.2 Creació i definició d'una aplicació Django

Un projecte de Django està format per aplicacions. Cada aplicació és un subdirector que conté els seus models, controladors i vistes.

Per a la creació d'una aplicació executarem la següent comanda per a crear la primera aplicació, `hotels`:

```
./manage.py startapp hotels
```

Dins del directori de `hotel_social_network` s'ha creat la següent carpeta:

```
hotels/
    models.py
    admin.py
    urls.py
    views.py
    tests.py
```

9.2.1 Models

En aquest fitxer definim el objectes de la nostra aplicació els quals necessitem guardar la seva informació a la base de dades.

Cada objecte representarà una taula de la base de dades i cada camp es correspon amb una columna de la taula. Un exemple seria:

```
class Hotel(models.Model):
    name = models.CharField(max_length=255)

class Review(models.Model):
    PUNCTUATIONS = (
        (1.0, "Awful"),
        (2.0, "Passable"),
        (3.0, "Good"),
        (4.0, "Cool"),
        (5.0, "Awesome")
    )
```

```
hotel = models.ForeignKey(Hotel)
title = models.CharField(max_length=255)
punctuation = models.IntegerField(choices=PUNTUATIONS)
```

Per a poder crear les taules de la base de dades a partir dels models de *Django* tenim les comandes *makemigrations* i *migrate*. Executem:

```
./manage.py makemigrations hotels
```

I la seva sortida és:

```
Migrations for 'hotels':
  0001_initial.py:
    - Create model Hotel
    - Create model Review
```

Això significa que ha creat el fitxer `0001_initial.py` que conté les comandes SQL per a la creació de dues taules, `hotel` i `review`. Tot seguit executem:

```
./manage.py migrate
```

La sortida de la comanda serà:

```
Operations to perform:
  Apply all migrations: admin, contenttypes, sessions, auth, hotels
Running migrations:
  Rendering model states... DONE
  Applying hotels.0001_initial... OK
```

Aquesta comanda ens ha aplicat l'script que contenia el fitxer `0001_initial.py` i ha creat les següents taules SQL:

hsn_hotel	
name	type
id	int(11)
name	varchar(255)

hsn_review	
id	int(11)
hotel	int(11)
title	varchar(255)
punctuation	float

9.2.2 Admin

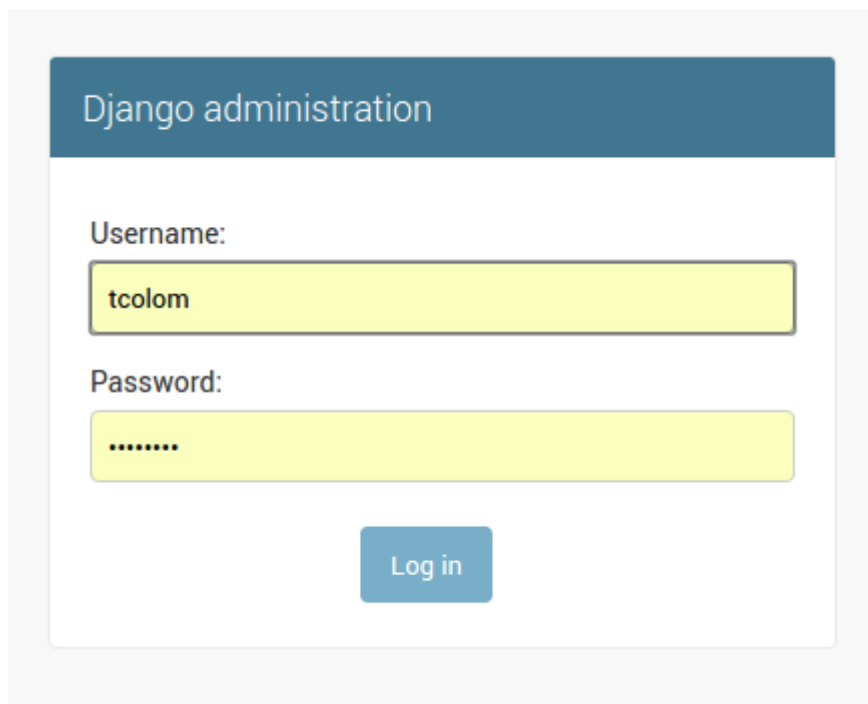
Django proporciona una interfície on podem gestionar el contingut de la base de dades. Per accedir-hi ho podem fer amb el propi servidor de Django. Per engegar-lo hem d'executar la comanda:

```
./manage.py runserver
```

La sortida es aquesta:

```
Performing system checks...
System check identified no issues (0 silenced).
May 08, 2016 - 20:09:53
Django version 1.9.6, using settings 'hotel_social_network.settings'
Starting development server at http://127.0.0.1:8000/
```

Accedint a la url <http://127.0.0.1:8000/admin/> ens trobem la pantalla de login del panel d'administració.



Per a crear un usuari i així poder accedir, hem d'executar la següent comanda:

```
./manage.py createsuperuser
```

Ens apareixerà un diàleg de consola per a introduir les nostres credencials:

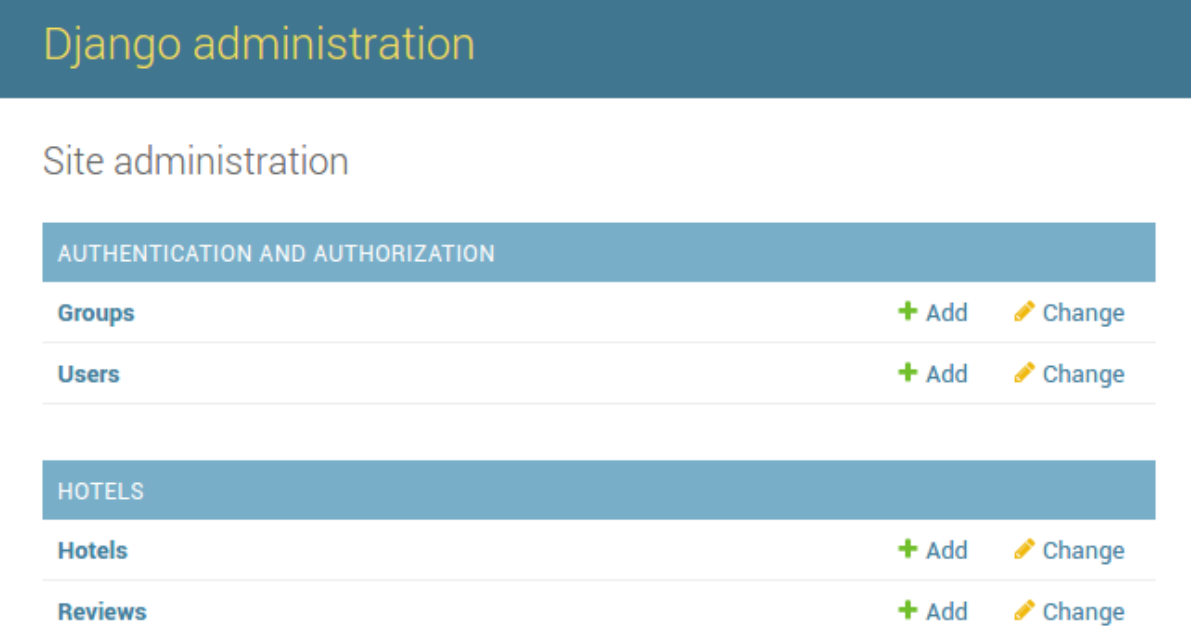
```
Username (leave blank to use 'tcolom'):
Email address: test@test.com
Password:
Password (again):
Superuser created successfully.
```

Si volem gestionar els models que hem creat anteriorment els hem d'enregistrar al panell d'administració, per això hem d'editar el fitxer *admin.py* i afegir-hi les següents línies.

```
from .models import Hotel, Review

admin.site.register(Hotel)
admin.site.register(Review)
```

Accedint un altre cop al panell d'administració, fent login ens apareix la següent pantalla:



Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

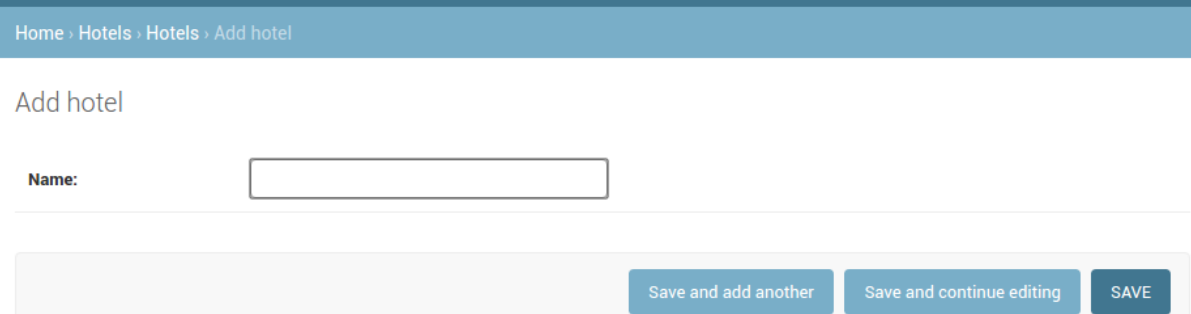
Groups	+ Add	Change
Users	+ Add	Change

HOTELS

Hotels	+ Add	Change
Reviews	+ Add	Change

Des de aquí podrem administrar els usuaris, els hotels i les opinions de la base de dades.

Si premem el botó *Add* de la secció hotels ens apareix el següent formulari per a la creació d'un hotel.



Home > Hotels > Hotels > Add hotel

Add hotel

Name:

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Emplenem els camps i li donem a *SAVE*. Això ens crearà una nova fila a la taula hotels de la base de dades amb els valors introduïts.

Si els volem editar podem accedir al llistat d'hotels del panell que té aquest aspecte.

Select hotel to change

ADD HOTEL +

Action: 0 of 1 selected☐ HOTEL☐ Test

1 hotel

9.2.3 URLs

Per a mostrar la informació a l'usuari final haurem de crear un punt d'entrada a la pàgina web, per això hem de definir les URLs. Primer accedim al fitxer *hotel_social_network/urls.py* i afegim la següent línia:

```
url(r'^hotels/', include('hotels.urls'))
```

el resultat serà el següent:

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^hotels/', include('hotels.urls'))
]
```

La primera línia afegeix totes les URLs necessàries per navegar pel panell d'administració sota el patró */admin/*. La segona afegeix totes les URLs definides a *hotels/urls.py* i s'accedirà sota el patró */hotels/*.

Tot seguit anem al fitxer *hotels/urls.py* i afegim el següent:

```
urlpatterns = [
    url(
        r'^$',
        HotelListView.as_view(),
        name='hotel-list'
    ),
    url(
        r'^(?P<pk>[0-9]+)$',
        HotelDetailView.as_view(),
        name='hotel-detail'
    )
]
```

Cada URL es defineix amb 3 paràmetres: El primer és l'expressió regular que ha de complir la URL del navegador per a coincidir amb aquesta entrada. El segon indica quina *View* s'ha d'executar i el tercer és el nom d'aquesta URL per a poder referenciar-la a altres parts del codi.

9.2.4 Views

Les *Views* són les funcions *Django* que s'encarreguen de rebre una petició web (*request*) i retornar una resposta web (*response*).

Django per defecte incorpora una sèrie de *Views* que ens faciliten la feina a l'hora de programar ja que fan la major part de la lògica. S'anomenen *Class Based Views* i l'únic que hem de fer és heretar-les i afegir o modificar els atributs o funcions que calguin.

Dos exemple serien *ListView* i *DetailView*. Per ambdues *Views*, només hem d'indicar a quin model fan referència i *Django* farà la resta per tal de convertir les peticions en resposta amb tota la informació relacionada amb el model indicat.

La *ListView* no necessita cap paràmetre d'entrada, i ens retorna la llista d'objectes definits a l'atribut model que tenim a la base de dades. El següent exemple retorna la llista d'hotels:

```
class HotelListView(ListView):  
    model = Hotel
```

La *DetailView* espera com a paràmetre d'entrada l'identificador del model per a poder mostrar tota la informació d'aquella instància. El següent exemple mostra la informació de l'Hotel amb *id* el paràmetre *pk* que s'ha passat per la URL amb nom *hotel-detail*:

```
class HotelDetailView(DetailView):  
    model = Hotel
```

9.2.5 Templates

Els *Templates* de *Django* s'encarreguen de la lògica de presentació on l'usuari realitza les interaccions amb l'aplicació.

Al directori *hotel_social_network/templates* crearem una carpeta de *Templates* per a cada aplicació. Per exemple, l'aplicació hotel tindria aquesta configuració:

```
hotel_social_network/templates/hotels  
    hotel_list.html  
    hotel_detail.html
```

Quan creem una *View* tipus *ListView* i definim el model, aquest va a buscar el *Template* corresponent a la carpeta

```
<nom de l'aplicació a que pertany el model>/<nom del model>_list.html
```

Sempre es pot sobreescriure passant el paràmetre a la classe *template_name*.

Per a poder dibuixar la llista d'hotels el contingut del *Template* haurà de ser:

```
{% for hotel in object_list %}  
    <a href="{% url 'hotel-detail' hotel.id %}">{{hotel.name}}</a>  
{% endfor %}
```


Consisteix en un bucle sobre la llista d'hotels. Per a cada un, creem un *link* en *HTML* el qual l'*anchor* és el nom de l'hotel i el *link* al que fa referència és la URL amb nom *hotel-detail* i l'hi passem com a paràmetre l'identificador de l'hotel.

El mateix passa amb un *View* tipus *DetailView*, el nom del *Template* ha de ser:

```
<nom de l'aplicació a que pertany el model>/<nom del model>_detail.html
```

El contingut de *hotel_detail.html* per a mostrar la informació d'un hotel en concret i les opinions que han fet els usuaris hauria de ser:

```
<h1>{{object.name}} {{object.category}}</h1>
<p>{{object.adress}}</p>

{% for review in object.review_set.all %}
    <h3>{{review.title}}</h3>
    <p>{{review.punctuation}} over 5</p>
{% endfor %}

<a href="{% url 'review-form' object.id %}">Create review</a>
```

Amb aquestes línies de codi mostrem la informació bàsica de l'hotel.

Tot seguit iterem sobre les opinions de l'hotel accedint a la relació per clau forana amb aquesta comanda *object.review_set.all* (*object* es refereix a l'hotel, *review_set* és refereix a les opinions que tenen com a hotel l'objecte hotel actual, *.all* es refereix a que es mostrin totes sense cap filtre).

El *link* final fa referència a una URL amb el formulari per a crear una nova opinió de l'hotel seleccionat.

9.2.6 Forms

Els formularis de *Django* són la interfície que ens permeten rebre dades d'entrada de l'usuari.

Hi ha dues maneres d'implementar-los, utilitzar un *Form* de *Django* i instànciar-lo a la vista o utilitzar la *CreateView* que porta implícit la declaració i la utilització del *Form*.

Al fitxer de *Views* afegim el següent codi:

```
class ReviewCreateView(CreateView):
    template_name = 'hotels/review_form.html'
    model = Review
    fields = ('hotel', 'title', 'punctuation')

    def get_success_url(self):
        return reverse('hotel-detail',kwargs=dict(pk=self.object.hotel.pk))
```

Aquesta *View* utilitzarà el *Template review_form.html*, el *Model Review* per a crear una nova instància de *Review* i els camps del formulari seran els *fields hotel, title i punctuation* del model *Review*.

Un cop el formulari s'hagi validat i les dades s'hagin guardat a la base de dades, la *View* redirigirà a l'usuari a la URL *hotel-detail* on podrà veure la *Review* que acaba de crear.

Al fitxer d'URLs s'hauria d'afegir la següent entrada per a poder accedir al formulari:

```
url(
    r'^(?P<pk>[0-9]+)/create-review$',
    ReviewCreateView.as_view(),
    name='review-form'
)
```

Un cop l'usuari ha accedit a la URL es mostrarà el formulari. El *Template* que renderitza el formulari conté el següent:

```
<form method="POST">
    {% csrf_token %}
    {{form}}
    <button>Save</button>
</form>
```

El tag *csrf_token* introdueix un valor ocult per evitar atacs tipus CSRF. El tag *form* itera sobre els *fields* definits a la vista i els rendereitza segons del tipus que siguin: Un *CharField* es mostra com a *input*, un *TextField* es mostra com a *textarea*, un *ForeignKey* es mostra com un *select*.

Un cop l'usuari fa click al botó *Save* s'envia la informació a la vista i executa la funció *is_valid()*. En el cas de ser vàlid executa la funció *form_valid()* i en cas contrari *form_invalid()*. La funció *form_valid()* guarda la informació a la base de dades i executa la funció *get_success_url()*. La funció *form_invalid()* retorna la informació introduïda al *Template* indicant els errors que s'han produït.

9.2.7 Tests

Finalment *Django* ens proporciona una interfície per a poder realitzar test unitaris i d'integració.

Dins de l'aplicació creem una carpeta anomenada *tests/* i dins d'aquest directori creem un fitxer per a testejar els *Models* anomenat *test_models.py* amb el següent contingut:

```
class HotelModelTest(TestCase):
    def test_create_hotel(self):
        hotel = Hotel.objects.create(name="test")
        self.assertEqual(hotel.name, "test")
        self.assertEqual(Hotel.objects.count(), 1)
        self.assertTrue(Hotel.objects.exists())
```

Amb aquest test provem la creació d'una instància *d'Hotel*. Un cop creat comprovem que s'ha creat amb el nom que toca, que només s'ha creat una instància i que la comanda `.exists()` retorna cert.

Podem crear un altre fitxer anomenat `test_views.py` en el qual testejarem les Views. El contingut serà el següent:

```
class HotelViewsTest(TestCase):
    def test_create_review(self):
        self.assertFalse(Review.objects.exists())
        hotel = Hotel.objects.create(name="test")
        data = {
            "hotel": hotel.id,
            "title": "test",
            "punctuation": 3
        }
        self.client.post(
            reverse_lazy("review-form", kwargs=dict(pk=hotel.pk)),
            data=data
        )
        self.assertTrue(Review.objects.exists())
        self.assertEqual(Review.objects.count(), 1)
        review = Review.objects.get()
        self.assertEqual(review.hotel, hotel)
        self.assertEqual(review.title, "test")
        self.assertEqual(review.punctuation, 3)
```

En aquest test, primer de tot, ens assegurem que no existeix cap *Review*, després fem la crida que faria el formulari de crear *Review*. Si la crida és correcte podem testejar que s'ha creat una *Review* amb els paràmetres que hem passat al formulari.

Finalment executarem els tests per veure que tot és correcte amb la següent comanda:

```
./manage.py test
```

La sortida hauria de ser:

```
Creating test database for alias 'default'...
```

```
..
```

```
-----
Ran 2 tests in 0.015s
```

```
OK
```

```
Destroying test database for alias 'default'...
```

Ens indica que s'ha creat una base de dades per test, que s'han executat 2 tests correctament i s'ha eliminat la base de dades de test.

9.3 Proves

Aquesta secció la dividirem en dos parts depenent del tipus de proves: les proves unitàries i les proves d'integració.

9.3.1 Proves unitàries

Les proves unitàries seran les responsables de testejar el codi font de l'aplicació i verificar que realitzen la funcionalitat esperada.

Sempre que sigui possible optarem per la metodologia *TDD (Test-Driven Development)*, és a dir, crear el test abans de realitzar la funcionalitat, veure que falla, i un cop feta la funcionalitat comprovar que el test és correcte.

Si es dona el cas en el que *TDD* no sigui possible, realitzarem *BDD (Behavior-Driven Development)*, realitzar els tests sobre una funcionalitat existent.

Per a totes les aplicacions del nostre projecte, testejarem les mateixes parts i farem les mateixes proves. No obstant, per alguna aplicació es poden fer test específics.

Als models testejarem que la creació s'ha fet correctament, validant el tipus d'instància, el nombre de files creades a la base de dades, els valors dels atributs de la instància i totes les funcions del model.

La part dels formularis testejarem el camps obligatoris i els que no, i les validacions de cada un dels camps.

Finalment farem proves sobre les vistes. En aquesta part verificarem quines d'elles requereixen un usuari logat i quines no, quines poden ser accessibles per determinats usuaris, els tipus de peticions HTTP que esperen i revisar les diferents respostes HTTP.

9.3.1.1 Usuaris

Prova	Resultat esperat
Models	
Creació d'una instància d'Usuari amb tots els camps correctes.	S'ha creat una i només una instància del modelUsuari amb tots els camps correctament.
Creació d'una instància d'Usuari sense algun dels camps obligatoris.	No es crea cap instància d'usuari i el sistema ens mostra quins són els camps que s'han d'introduir.
Creació d'una instància d'Usuari amb algun camp incorrecte (strings massa llargs, dates de naixement inexistents, ...).	No es crea cap instància d'usuari i el sistema ens mostra quins són els errors que s'han introduït.
A partir d'una instància d'usuari obtenir l'edat.	Ens retornarà l'edat de l'usuari a partir de la seva data de naixement.
A partir d'una instància d'usuari obtenir els posts que ha publicat.	Ens retornarà la llista de posts que ha creat un usuari.
Forms	
Enviar el formulari de login d'Usuari amb tots els camps buits.	Ens retornarà un error dient quins són els camps obligatoris
Enviar el formulari de login d'Usuari amb dades incorrectes.	Ens retornarà un error dient quins són els camps incorrectes.
Enviar el formulari de login amb dades d'un usuari inexistent.	Ens retornarà un error informant que l'usuari no existeix o que les dades que s'han introduït són incorrectes.
Enviar el formulari de login amb dades d'un usuari inactiu.	Ens retornarà un error informant que l'usuari no existeix o que les dades que s'han introduït són incorrectes.
Enviar el formulari de login amb dades correctes d'un usuari.	No retorna cap error i la instància de l'usuari està identificada en el sistema.
Enviar el formulari de registre d'Usuari amb tots els camps buits.	Ens retornarà un error dient quins són els camps obligatoris.
Enviar el formulari de registre d'Usuari amb dades incorrectes.	Ens retornarà un error dient quins són els camps incorrectes.
Enviar el formulari de registre amb dades d'un usuari ja existent.	Ens retornarà un error informant de que ja existeix un usuari amb aquelles dades.
Enviar el formulari de registre amb dades valides per un nou usuari.	No ens retorna cap error i s'ha creat una nova instància d'Usuari a la base de dades.
Enviar el formulari d'edició de perfil amb tots els camps buits.	Ens retornarà un error dient quins són els camps obligatoris.

Enviar el formulari d'edició de perfil amb dades incorrectes.	Ens retornarà un error dient quins són els camps incorrectes.
Enviar el formulari d'edició de perfil amb dades correctes.	No ens retorna cap error i la instància de l'usuari està modificada amb les dades introduïdes.
Views	
Petició amb un usuari logat a la vista de login.	Retorna una redirecció cap a la pàgina principal.
Petició GET sense usuari a la vista de login.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari incorrectes a la vista de login.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari correctes a la vista de login.	Retorna una redirecció a la pàgina principal.
Petició amb un usuari logat a la vista de registre.	Retorna una redirecció cap a la pàgina principal.
Petició GET sense usuari a la vista de registre.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari incorrectes a la vista de registre.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari correctes a la vista de registre.	Retorna una redirecció a la pàgina del perfil de l'usuari.
Petició GET amb un usuari no logat a la vista de perfil.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista de perfil.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista de perfil.	Retorna un error indicant que no admet peticions tipus POST.
Petició amb un usuari no logat a la vista d'edició de perfil.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista d'edició de perfil.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari incorrectes a la vista d'edició de perfil.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari correctes a la vista d'edició de perfil.	Retorna una redirecció a la pàgina del perfil de l'usuari.
Petició GET amb un usuari no logat a la vista llistat d'usuaris.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista llistat d'usuaris.	Retorna una resposta HTTP amb un codi 200.

Petició POST a la vista llistat d'usuaris.	Retorna un error indicant que no admet peticions tipus POST.
---	--

9.3.1.2 Hotels

Prova	Resultat esperat
Models	
Creació d'una instància d'Hotel amb tots els camps correctes.	S'ha creat una i només una instància del model Hotel amb tots els camps correctament.
Creació d'una instància d'Hotel sense algun dels camps obligatoris.	No es crea cap instància d'hotel i el sistema ens mostra quins són els camps que s'han d'introduir.
Creació d'una instància d'Hotel amb algun camp incorrecte.	No es crea cap instància d'hotel i el sistema ens mostra quins són els errors que s'han introduït.
A partir d'una instància d'hotel obtenir els usuaris que hi ha allotjats.	Ens retornarà la llista d'usuaris que tenen el seu camp hotel aquesta instància d'hotel.
Forms	
Per aquest model no cal fer tests unitaris dels formularis ja que es fan pel panell d'administració i Django els proporciona.	
Views	
Petició GET amb un usuari no logat a la vista llistat d'hotels.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista llistat d'hotels.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista llistat d'hotels.	Retorna un error indicant que no admet peticions tipus POST.
Petició GET amb un usuari no logat a la vista fitxa d'hotels.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista fitxa d'hotels.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista fitxa d'hotels.	Retorna un error indicant que no admet peticions tipus POST.

9.3.1.3 Mur

Prova	Resultat esperat
Models	
Creació d'una instància de Post amb tots els camps correctes.	S'ha creat una i només una instància del model Post amb tots els camps correctament.
Creació d'una instància de Post sense algun	No es crea cap instància de post i el sistema ens

dels camps obligatoris.	mostra quins són els camps que s'han d'introduir.
Creació d'una instància de Post amb algun camp incorrecte.	No es crea cap instància de post i el sistema ens mostra quins són els errors que s'han introduït.
A partir d'una instància de post obtenir els seus comentaris.	Ens retornarà la llista de comentaris que tenen en el camp post aquesta instància de post.
Creació d'una instància de Comentari amb tots els camps correctes.	S'ha creat una i només una instància del model Comentari amb tots els camps correctament.
Creació d'una instància de Comentari sense algun dels camps obligatoris.	No es crea cap instància de comentari i el sistema ens mostra quins són els camps que s'han d'introduir.
Creació d'una instància de Comentari amb algun camp incorrecte.	No es crea cap instància de comentari i el sistema ens mostra quins són els errors que s'han introduït.
Forms	
Enviar el formulari de creació de Post sense camps	Ens retornarà un error dient quins són els camps obligatoris.
Enviar el formulari de creació de Post amb dades incorrectes.	Ens retornarà un error dient quins són els camps incorrectes.
Enviar el formulari creació de Post amb dades valides.	No ens retorna cap error i s'ha creat una nova instància de Post a la base de dades.
Enviar el formulari de edició de Post sense camps	Ens retornarà un error dient quins són els camps obligatoris.
Enviar el formulari de edició de Post amb dades incorrectes.	Ens retornarà un error dient quins són els camps incorrectes.
Enviar el formulari edició de Post amb dades valides.	No ens retorna cap error i s'ha modificat la instància de Post a la base de dades.
Enviar el formulari de creació de Comentari sense camps	Ens retornarà un error dient quins són els camps obligatoris.
Enviar el formulari de creació de Comentari amb dades incorrectes.	Ens retornarà un error dient quins són els camps incorrectes.
Enviar el formulari creació de Comentari amb dades valides.	No ens retorna cap error i s'ha creat una nova instància de Comentari a la base de dades.
Enviar el formulari de edició de Comentari sense camps	Ens retornarà un error dient quins són els camps obligatoris.
Enviar el formulari de edició de Comentari amb dades incorrectes.	Ens retornarà un error dient quins són els camps incorrectes.
Enviar el formulari edició de Comentari amb	No ens retorna cap error i s'ha modificat la

dades valides.	instància de Comentari a la base de dades.
Views	
Petició GET amb un usuari no logat a la vista llistat de posts.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista llistat de posts.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista llistat de posts.	Retorna un error indicant que no admet peticions tipus POST.
Petició GET amb un usuari no logat a la vista de post.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista de post.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista fitxa de post.	Retorna un error indicant que no admet peticions tipus POST.
Petició amb un usuari no logat a la vista d'edició de post.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista d'edició de post.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari incorrectes a la vista d'edició de post.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari correctes a la vista d'edició de post.	Retorna una redirecció a la pàgina del post.
Petició amb un usuari no logat a la vista d'eliminació de post.	Retorna una redirecció cap a la pàgina de login.
Petició no DELETE amb un usuari logat a la vista d'eliminació de post.	Retorna un error indicant que només admet peticions tipus DELETE.
Petició DELETE a la vista d'eliminació de post.	Retorna una redirecció al llistat de posts.
Petició POST amb dades del formulari incorrectes a la vista d'edició de comentari.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari correctes a la vista d'edició de comentari.	Retorna una redirecció a la pàgina del post.
Petició amb un usuari no logat a la vista d'eliminació de comentari.	Retorna una redirecció cap a la pàgina de login.
Petició no DELETE amb un usuari logat a la vista d'eliminació de comentari.	Retorna un error indicant que només admet peticions tipus DELETE.
Petició DELETE a la vista d'eliminació de	Retorna una redirecció a la pàgina de post.

comentari.	
------------	--

9.3.1.4 Opinions

Prova	Resultat esperat
Models	
Creació d'una instància d'Opinió amb tots els camps correctes.	S'ha creat una i només una instància del model Opinió amb tots els camps correctament.
Creació d'una instància d'Opinió sense algun dels camps obligatoris.	No es crea cap instància d'opinió i el sistema ens mostra quins són els camps que s'han d'introduir.
Creació d'una instància d'Opinió amb algun camp incorrecte.	No es crea cap instància d'opinió i el sistema ens mostra quins són els errors que s'han introduït.
A partir d'una instància d'Opinió actualitzar la puntuació.	Ens retornarà el valor de la mitjana de les puntuacions de les opinions.
Forms	
Enviar el formulari de creació d'Opinió sense camps	Ens retornarà un error dient quins són els camps obligatoris.
Enviar el formulari de creació d'Opinió amb dades incorrectes.	Ens retornarà un error dient quins són els camps incorrectes.
Enviar el formulari creació d'Opinió amb dades vàlides.	No ens retorna cap error i s'ha creat una nova instància d'Opinió a la base de dades.
Views	
Petició GET amb un usuari no logat a la vista llistat d'hotels.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista llistat d'hotels.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista llistat d'hotels.	Retorna un error indicant que no admet peticions tipus POST.
Petició amb un usuari logat a la vista de creació d'Opinió.	Retorna una redirecció cap a la pàgina principal.
Petició GET sense usuari a la vista de creació d'Opinió.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari incorrectes a la vista de creació d'Opinió.	Retorna un error indicant quins són els camps incorrectes.
Petició POST amb dades del formulari correctes a la vista de creació d'Opinió..	Retorna una redirecció al llistat d'opinions.

9.3.1.5 Esdeveniments

Prova	Resultat esperat
Models	
Creació d'una instància d'Esdeveniment amb tots els camps correctes.	S'ha creat una i només una instància del model Esdeveniment amb tots els camps correctament.
Creació d'una instància d'Esdeveniment sense algun dels camps obligatoris.	No es crea cap instància d'opinió i el sistema ens mostra quins són els camps que s'han d'introduir.
Creació d'una instància d'Esdeveniment amb algun camp incorrecte.	No es crea cap instància d'esdeveniment i el sistema ens mostra quins són els errors que s'han introduït.
Forms	
Enviar el formulari de edició d'Esdeveniment sense camps	Ens retornarà un error dient quins són els camps obligatoris.
Enviar el formulari de edició d'Esdeveniment amb dades incorrectes.	Ens retornarà un error dient quins són els camps incorrectes.
Enviar el formulari edició d'Esdeveniment amb dades valides.	No ens retorna cap error i s'ha modificat la instància d'Esdeveniment a la base de dades.
Enviar el formulari de creació d'Esdeveniment sense camps	Ens retornarà un error dient quins són els camps obligatoris.
Enviar el formulari de creació d'Esdeveniment amb dades incorrectes.	Ens retornarà un error dient quins són els camps incorrectes.
Enviar el formulari creació d'Esdeveniment amb dades valides.	No ens retorna cap error i s'ha creat una nova instància d'Esdeveniment a la base de dades.
Views	
Petició GET amb un usuari no logat a la vista llistat d'esdeveniments.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista llistat d'esdeveniments.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista llistat d'esdeveniments.	Retorna un error indicant que no admet peticions tipus POST.
Petició GET amb un usuari no logat a la vista d'esdeveniment.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista d'esdeveniment.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista fitxa d'esdeveniment.	Retorna un error indicant que no admet peticions tipus POST.

Petició amb un usuari logat a la vista de creació d'esdeveniment.	Retorna una redirecció cap a la pàgina principal.
Petició GET sense usuari a la vista de creació d'esdeveniment.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari incorrectes a la vista de creació d'esdeveniment.	Retorna un error indicant quins són els camps incorrectes.
Petició POST amb dades del formulari correctes a la vista de creació d'esdeveniment.	Retorna una redirecció al llistat d'esdeveniments.
Petició amb un usuari no logat a la vista d'edició d'esdeveniment.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista d'edició d'esdeveniment.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari incorrectes a la vista d'edició d'esdeveniment.	Retorna una resposta HTTP amb un codi 200.
Petició POST amb dades del formulari correctes a la vista d'edició d'esdeveniment.	Retorna una redirecció a la pàgina del esdeveniment.
Petició amb un usuari no logat a la vista d'eliminació d'esdeveniment.	Retorna una redirecció cap a la pàgina de login.
Petició no DELETE amb un usuari logat a la vista d'eliminació d'esdeveniment.	Retorna un error indicant que només admet peticions tipus DELETE.
Petició DELETE a la vista d'eliminació d'esdeveniment.	Retorna una redirecció al llistat d'esdeveniments.

9.3.1.6 Anuncis

Prova	Resultat esperat
Models	
Creació d'una instància d'Anunci amb tots els camps correctes.	S'ha creat una i només una instància del model Anunci amb tots els camps correctament.
Creació d'una instància d'Anunci sense algun dels camps obligatoris.	No es crea cap instància d'Anunci i el sistema ens mostra quins són els camps que s'han d'introduir.
Creació d'una instància d'Anunci amb algun camp incorrecte.	No es crea cap instància d'Anunci i el sistema ens mostra quins són els errors que s'han introduït.
Forms	
Per aquest model no cal fer tests unitaris dels formularis ja que es fan pel panell d'administració i Django els proporciona.	

Views	
Petició GET amb un usuari no logat a la vista llistat d'Anuncis.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista llistat d'Anuncis.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista llistat d'Anuncis.	Retorna un error indicant que no admet peticions tipus POST.

9.3.1.7 Missatges

Prova	Resultat esperat
Models	
Creació d'una instància de Xat amb tots els camps correctes.	S'ha creat una i només una instància del model Xat amb tots els camps correctament.
Creació d'una instància de Xat sense algun dels camps obligatoris.	No es crea cap instància de Xat i el sistema ens mostra quins són els camps que s'han d'introduir.
Creació d'una instància de Xat amb algun camp incorrecte.	No es crea cap instància de Xat i el sistema ens mostra quins són els errors que s'han introduït.
Creació d'una instància de Missatge amb tots els camps correctes.	S'ha creat una i només una instància del model Missatge amb tots els camps correctament.
Creació d'una instància de Missatge sense algun dels camps obligatoris.	No es crea cap instància de Missatge i el sistema ens mostra quins són els camps que s'han d'introduir.
Creació d'una instància de Missatge amb algun camp incorrecte.	No es crea cap instància de Missatge i el sistema ens mostra quins són els errors que s'han introduït.
Forms	
Per aquest model no cal fer tests unitaris dels formularis ja que es fan pel panell d'administració i Django els proporciona.	
Views	
Petició GET amb un usuari no logat a la vista llistat de Xats.	Retorna una redirecció cap a la pàgina de login.
Petició GET amb un usuari logat a la vista llistat de Xats.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista llistat de Xats.	Retorna un error indicant que no admet peticions tipus POST.
Petició GET amb un usuari no logat a la vista de Xat.	Retorna una redirecció cap a la pàgina de login.

Petició GET amb un usuari logat a la vista de Xat.	Retorna una resposta HTTP amb un codi 200.
Petició POST a la vista de Xat.	Retorna un error indicant que no admet peticions tipus POST.
Petició amb un usuari no logat a la vista d'eliminació de xat.	Retorna una redirecció cap a la pàgina de login.
Petició no DELETE amb un usuari logat a la vista d'eliminació de xat.	Retorna un error indicant que només admet peticions tipus DELETE.
Petició DELETE a la vista d'eliminació de xat.	Retorna una redirecció al llistat de xats.
Petició amb un usuari no logat a la vista d'eliminació de missatge.	Retorna una redirecció cap a la pàgina de login.
Petició no DELETE amb un usuari logat a la vista d'eliminació de missatge.	Retorna un error indicant que només admet peticions tipus DELETE.
Petició DELETE a la vista d'eliminació de missatge.	Retorna una redirecció al xat.

9.3.2 Proves d'integració

Les proves d'integració seran les encarregades de verificar que la unió de totes les peces funciona correctament, en el nostre cas, les realitzarem sobre la part visual interactuant amb el navegador.

9.3.2.1 Usuaris

Prova	Resultat esperat
Formulari de login	
L'usuari entra a la pàgina de login	Es mostra un formulari amb els camps email i contrasenya i un botó per enviar el formulari.
L'usuari omple el formulari deixant algun camp obligatori sense omplenar i li dona al botó "Enviar".	Apareixen missatges de "Camp obligatori" en vermell sota els camps que són obligatoris i s'han deixat buits.
L'usuari introdueix camps incorrectes (p.e: email no vàlid) en el formulari i li dona al botó "Enviar".	Apareixen missatges en vermell sota els camps que són incorrectes indicant l'error.
L'usuari omple el formulari correctament i li dona al botó "Enviar".	L'usuari és redirigit a la pàgina principal.
Formulari de registre	
L'usuari entra a la pàgina del formulari de registre.	Es mostra un formulari amb els camps i un botó per enviar el formulari.
L'usuari omple el formulari deixant algun	Apareixen missatges de "Camp obligatori" en

camp obligatori sense emplenar i li dóna al botó “Enviar”.	vermell sota els camps que són obligatoris i s’han deixat buits.
L’usuari introdueix camps incorrectes (p.e: email no vàlid) en el formulari i li dóna al botó “Enviar”.	Apareixen missatges en vermell sota els camps que són incorrectes indicant l’error.
L’usuari emplena el formulari correctament i li dóna al botó “Enviar”.	L’usuari és redirigit a la pàgina del seu perfil.
Perfil d’usuari	
L’usuari entra al perfil d’un usuari.	Es mostra el nom, la foto, l’edat, el sexe, la descripció (si en té) i els posts que ha publicat (si en té).
Formulari editar perfil	
L’usuari entra a la pàgina del formulari d’edició de perfil.	Es mostra un formulari amb els camps i un botó per enviar el formulari.
L’usuari emplena el formulari deixant algun camp obligatori sense emplenar i li dóna al botó “Enviar”.	Apareixen missatges de “Camp obligatori” en vermell sota els camps que són obligatoris i s’han deixat buits.
L’usuari introdueix camps incorrectes (p.e: descripció massa llarga) en el formulari i li dóna al botó “Enviar”.	Apareixen missatges en vermell sota els camps que són incorrectes indicant l’error.
L’usuari emplena el formulari correctament i li dóna al botó “Enviar”.	L’usuari és redirigit a la pàgina del seu perfil.
Llistat d’usuaris	
L’usuari entra al llistat d’usuaris.	Es mostren els usuaris de la base de dades amb el seu nom, edat, sexe i foto.

9.3.2.2 Hotels

Prova	Resultat esperat
Llistat d’hotels	
L’usuari entra al llistat d’hotels.	Es mostren els hotels de la base de dades amb el seu nom, categoria i imatge.
Fitxa d’hotel	
L’usuari entra a la fitxa d’un hotel.	Es mostra el títol, la categoria, l’adreça, el mapa amb la situació, la galeria d’imatges i les opinions dels usuaris.

9.3.2.3 Mur

Prova	Resultat esperat
-------	------------------

Llistat de posts	
L'usuari entra al llistat de posts.	Es mostren els posts de la base de dades amb el seu títol, comentari , imatge (si en té), usuari que l'ha publicat i comentaris (si en té).
Pàgina de post	
L'usuari entra a la pàgina de post.	Es mostra el títol del post, el contingut, la imatge (si en té), els comentaris (si en té) i el nom de l'usuari que l'ha fet.
Editar post	
L'usuari entra a la pàgina del formulari d'editar post.	Es mostra un formulari amb els camps i un botó per enviar el formulari.
L'usuari emplena el formulari deixant algun camp obligatori sense emplenar i li dóna al botó "Enviar".	Apareixen missatges de "Camp obligatori" en vermell sota els camps que són obligatoris i s'han deixat buits.
L'usuari introdueix camps incorrectes (p.e: email no vàlid) en el formulari i li dóna al botó "Enviar".	Apareixen missatges en vermell sota els camps que són incorrectes indicant l'error.
L'usuari emplena el formulari correctament i li dóna al botó "Enviar".	L'usuari és redirigit a la pàgina del post.
Eliminar post	
L'usuari prem el botó d'eliminar post.	Es redirigit a la pàgina d'eliminar post on hi apareix un missatge de confirmació amb dos botons, "Eliminar" o "Cancel·lar"
L'usuari prem el botó de "Cancel·lar".	És redirigit a la pàgina del post.
L'usuari prem el botó de "Eliminar".	És redirigit al llistat de posts i aquest ja no apareix.
Formulari crear comentari	
L'usuari emplena el formulari deixant el camp comentari sense emplenar i li dóna al botó "Enviar".	Apareix un missatge de "Camp obligatori" en vermell sota el camp comentari.
L'usuari introdueix un comentari massa llarg i li dóna al botó "Enviar".	Apareix un missatge en vermell sota el camp indicant l'error.
L'usuari emplena el formulari correctament i li dóna al botó "Enviar".	L'usuari és redirigit a la pàgina del post.
Editar comentari	
L'usuari entra a la pàgina del formulari de registre.	Es mostra un formulari amb els camps i un botó per enviar el formulari.
L'usuari emplena el formulari deixant algun camp obligatori sense emplenar i li dóna al botó "Enviar".	Apareixen missatges de "Camp obligatori" en vermell sota els camps que són obligatoris i s'han deixat buits.

L'usuari introdueix camps incorrectes (p.e: comentari massa llarg) en el formulari i li dóna al botó "Enviar".	Apareixen missatges en vermell sota els camps que són incorrectes indicant l'error.
L'usuari emplena el formulari correctament i li dóna al botó "Enviar".	L'usuari és redirigit a la pàgina de post.
Eliminar comentari	
L'usuari prem el botó d'eliminar comentari.	És redirigit a la pàgina d'eliminar comentari on hi apareix un missatge de confirmació amb dos botons, "Eliminar" o "Cancel·lar"
L'usuari prem el botó de "Cancel·lar".	És redirigit a la pàgina del post.
L'usuari prem el botó de "Eliminar".	És redirigit a la pàgina del post i el seu comentari ja no apareix.

9.3.2.4 Opinions

Prova	Resultat esperat
Llistat d'opinions	
L'usuari entra al llistat d'opinions d'un hotel.	Es mostren les opinions de l'hotel amb el seu títol, comentari positiu, comentari negatiu, resposta de l'hotel i les valoracions de cada servei en forma d'estrelles i el nom de l'usuari que l'ha realitzada.
Formulari de creació d'opinió	
L'usuari emplena el formulari deixant algun camp obligatori sense emplenar i li dóna al botó "Enviar".	Apareixen missatges de "Camp obligatori" en vermell sota els camps que són obligatoris i s'han deixat buits.
L'usuari introdueix camps incorrectes (p.e: comentari positiu massa llarg) en el formulari i li dóna al botó "Enviar".	Apareixen missatges en vermell sota els camps que són incorrectes indicant l'error.
L'usuari emplena el formulari correctament i li dóna al botó "Enviar".	L'usuari és redirigit al llistat d'opinions de l'hotel i la seva apareix la primera.

9.3.2.5 Esdeveniments

Prova	Resultat esperat
Llistat esdeveniments	
L'usuari entra al llistat d'esdeveniments.	Es mostren els esdeveniments de la base de dades amb el seu títol, descripció i data.
Formulari creació d'esdeveniment	
L'usuari entra a la pàgina del formulari de	Es mostra un formulari amb els camps i un botó

creació d'esdeveniment.	per enviar el formulari.
L'usuari emplena el formulari deixant algun camp obligatori sense emplenar i li dóna al botó "Enviar".	Apareixen missatges de "Camp obligatori" en vermell sota els camps que són obligatoris i s'han deixat buits.
L'usuari introdueix camps incorrectes (p.e: descripció massa llarga) en el formulari i li dóna al botó "Enviar".	Apareixen missatges en vermell sota els camps que són incorrectes indicant l'error.
L'usuari emplena el formulari correctament i li dóna al botó "Enviar".	L'usuari és redirigit al la fitxa de l'esdeveniment que acaba de crear.
Pàgina de l'esdeveniment	
L'usuari entra a la pàgina de l'esdeveniment.	Es mostra el títol, la descripció, la imatge (si en té), la data i el nom de l'usuari que l'ha fet.
Editar esdeveniment	
L'usuari entra a la pàgina del formulari de creació d'esdeveniment.	Es mostra un formulari amb els camps i un botó per enviar el formulari.
L'usuari emplena el formulari deixant algun camp obligatori sense emplenar i li dóna al botó "Enviar".	Apareixen missatges de "Camp obligatori" en vermell sota els camps que són obligatoris i s'han deixat buits.
L'usuari introdueix camps incorrectes (p.e: descripció massa llarga) en el formulari i li dóna al botó "Enviar".	Apareixen missatges en vermell sota els camps que són incorrectes indicant l'error.
L'usuari emplena el formulari correctament i li dóna al botó "Enviar".	L'usuari és redirigit al la fitxa de l'esdeveniment que acaba de crear.
Eliminar esdeveniment	
L'usuari prem el botó d'eliminar esdeveniment.	És redirigit a la pàgina d'eliminar esdeveniment on hi apareix un missatge de confirmació amb dos botons, "Eliminar" o "Cancel·lar"
L'usuari prem el botó de "Cancel·lar".	És redirigit a la pàgina del post.
L'usuari prem el botó de "Eliminar".	És redirigit al llistat de posts i aquest ja no apareix.

9.3.2.6 Anuncis

Prova	Resultat esperat
Llistat d'anuncis	
L'usuari entra al llistat d'anuncis.	Es mostren els anuncis de la base de dades amb el seu títol i imatge.
L'usuari entra al llistat sense anuncis.	Es mostra un missatge informant que no hi ha anuncis.

9.3.2.7 Missatges

Prova	Resultat esperat
Llistat Xats	
L'usuari entra al llistat de xats.	Es mostren els xats de la base de dades de l'usuari que accedeix amb altres usuaris.
Xat	
L'usuari entra a la pàgina de xat	Apareixen tots els missatges intercanviats entre l'usuari i el receptor per orde cronològic.
Eliminar xat	
L'usuari prem el botó d'eliminar xat.	És redirigit a la pàgina d'eliminar xat on hi apareix un missatge de confirmació amb dos botons, "Eliminar" o "Cancel·lar"
L'usuari prem el botó de "Cancel·lar".	És redirigit al llistat de xats.
L'usuari prem el botó de "Eliminar".	És redirigit al llistat de xats i aquest ja no apareix.
Eliminar missatge	
L'usuari prem el botó d'eliminar missatge.	És redirigit a la pàgina d'eliminar missatge on hi apareix un missatge de confirmació amb dos botons, "Eliminar" o "Cancel·lar"
L'usuari prem el botó de "Cancel·lar".	És redirigit al xat.
L'usuari prem el botó de "Eliminar".	És redirigit al xat i aquest ja no apareix.

10 Conclusió

Feia molt temps que volia implementar aquesta idea i per fi he trobat la oportunitat per implementar-la correctament. He pogut redactar una memòria bastant completa y desenvolupar totes les funcionalitats definides a l'abast.

A nivell personal he quedat molt satisfet amb els resultats i em quedo a l'espera de veure l'acceptació general del públic.

A continuació detallaré les desviacions i les limitacions que m'he trobat del projecte, els coneixements que he adquirit a la carrera per fer-lo i finalment els plans de futur que tinc pensats per aquest.

10.1 Desviacions en la planificació i pressupost

El projecte es dona per acabat el dia 31 d'Agost del 2016 amb una suma total de 546 hores.

A l'apartat 4.1 Control de gestió varem fer una estimació de les hores que es dedicarien en desenvolupar aquest projecte. Un cop finalitzat el resultat és el següent.

Recurs	Hores estimades	Hores dedicades	Desviació	Preu/hora	Preu estimat	Preu final
Cap de projecte	70h	76h	+6h	30€	2.100€	2.280€
Analista	140h	158h	+18h	20€	2.800€	3.160€
Dissenyador	80h	120h	+40h	20€	1.600€	2.400€
Programador	180h	192h	+12h	15€	2.700€	2.880€
Total	470h	546h	76h		9.200€	10.720€

Com es pot observar s'ha produït un desviament de 76h sobre l'estimació inicial. Això és aproximadament un 16% sobre el pressupost inicial.

Es tracta d'una desviació una mica per sobre del risc del 15% que es va acordar al inici.

Concepte	Preu
Recursos humans	10.720€
Software	69€
Hardware	452€
Costos generals	562€
Total	11.803€

Finalment el cost total del projecte puja a 11.803€, uns 1.520€ més car que la previsió inicial que eren 10.283€, degut a l'augment d'hores dedicades per part de l'equip.

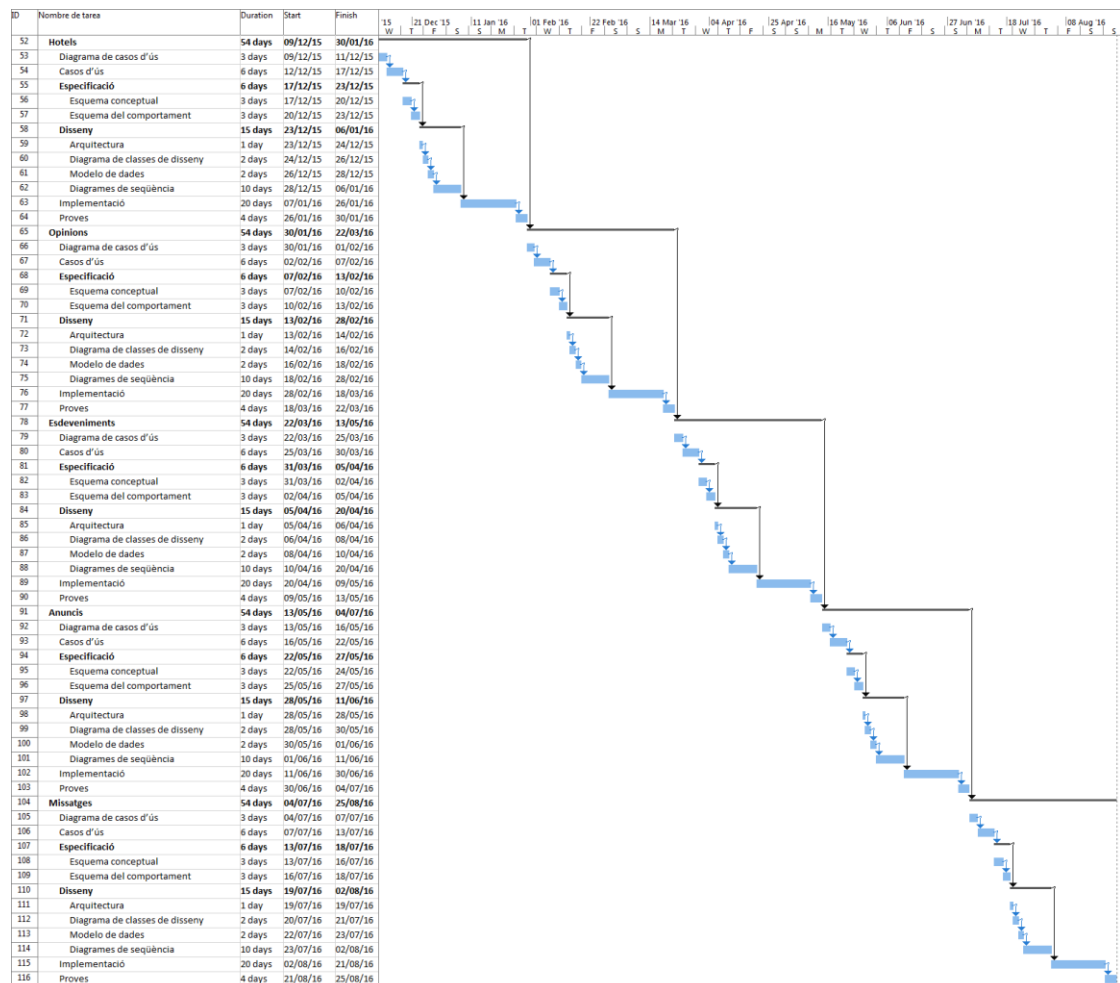
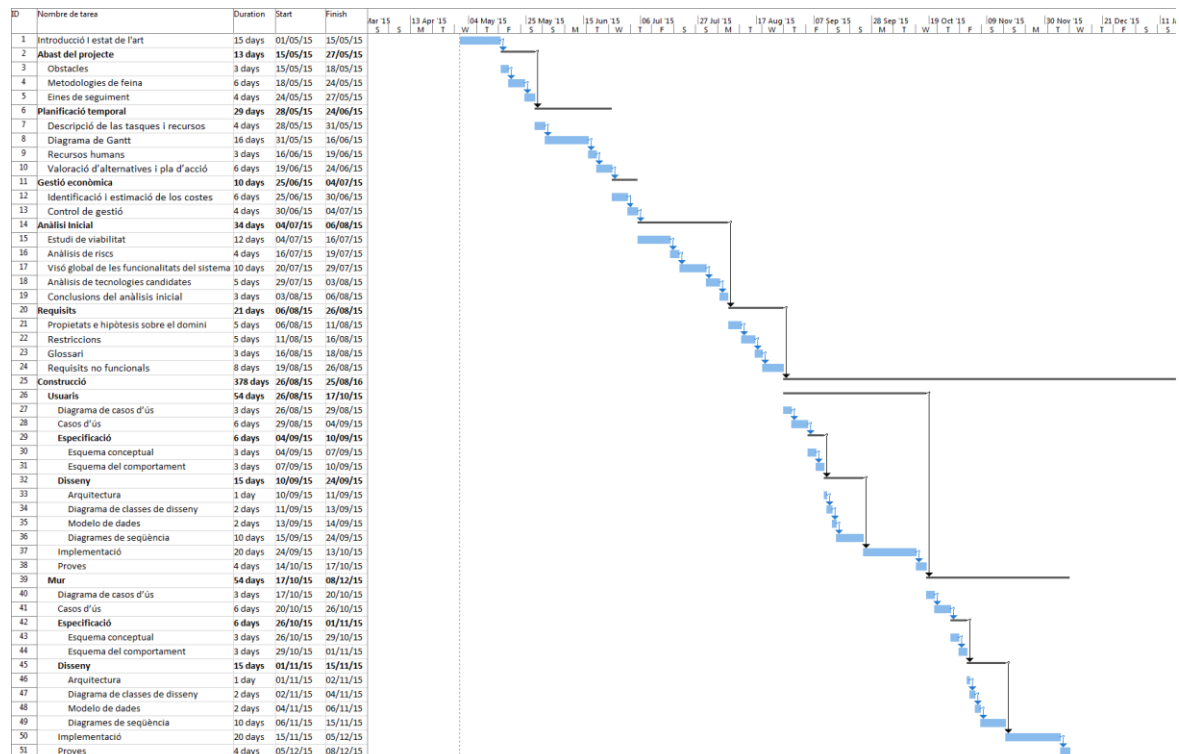
Dels possibles riscos que ens podríem trobar durant la realització del projecte ens hem trobat els següents:

Desviament en la planificació d'alguna de les fases. Per tal de reduir les conseqüències d'aquest risc varem decidir simplificar la funcionalitat del xat ja que comportava molt de temps de desenvolupament.

Desconeixement de la tecnologia. Degut a al risc anterior, es va decidir simplificar la funcionalitat del xat ja que requeria l'estudi i la implementació d'una tecnologia desconeguda com son les cues, en el nostre cas amb RabbitMQ.

A continuació mostrem el diagrama de Gantt definitiu amb les modificacions pertinents degut a les desviacions.

10.2 Diagrama de Gantt final



10.3 Limitacions i dificultats

El principal problema que m'he trobat ha estat el temps, ja que combinar una feina a jornada completa amb la realització d'aquest projecte al mateix temps, no ha estat fàcil.

La part que s'ha vist més afectada pel temps ha estat la programació. He hagut de prendre algunes decisions per fer les funcionalitats el més simple possible. El resultat ha estat tenir un xat entre usuaris molt bàsic, a forma de demostració. No utilitza cap sistema de Web Sockets com tenia previst.

Una de les dificultats que em vaig trobar va ser a l'hora de mostrar el mur al l'usuari. El mur és un llistat format per posts, esdeveniments i anuncis ordenats cronològicament. El problema quan volem obtenir tots els elements és que s'han de fer tres peticions diferents a la base de dades i després ordenar els elements. Per solucionar-ho vaig fer una taula on guarda la informació comuna de les tres taules (post, esdeveniment i anunci), així només cal fer una *query* en aquella taula ja que els elements vindran ordenats.

Tot i així, l'objectiu de tenir un MVP (Minimum Viable Product) s'ha complert. He aconseguit tenir un prototip bàsic funcional, que es pugui anar provant i sobre els primers resultats o necessitats poder anar ampliant i millorant les funcionalitats.

Per altre banda, hi ha requisits no funcionals que no he pogut aconseguir per falta de temps o perquè suposaven un cost elevat.

Degut a que AWS no és gratuït i que realitzar proves d'autoescalat implicaria dedicar molt temps, no s'han pogut complir els següents requisits no funcionals:

- RNF-03 El sistema haurà de tenir capacitat per emmagatzemar dades segons la demanda.
- RNF-08 El sistema haurà de ser capaç d'atendre sol·licituds segons la demanda.
- RNF-10 El sistema realitzarà còpies de seguretat constantment.
- RNF-15 Recuperació de desastres.
- RNF-19 Totes les peticions i respostes del sistema hauran d'estar encriptades.

Per falta de temps també s'han deixat de banda la realització de documents o informes mencionats als següents requisits. Alguns d'ells també implicaven la acció d'una consultora externa o la intervenció d'un advocat:

- RNF-05 El sistema ha de tenir un procediment documentat de desplegament de l'aplicació tant per a servidors de desenvolupament com de producció.
- RNF-13 El sistema informarà als usuaris de la política de privacitat sobre les dades que disposa.
- RNF-14 Totes les dades seran tractades segons estableix la LOPD i la directiva de protecció de dades 95/46 de la UE.
- RNF-21 El sistema disposarà d'una secció d'ajuda.

Aquests requisits s'aniran complint segons es vagi desenvolupant l'aplicació en iteracions futures.

10.4 Coneixements aplicats adquirits durant la carrera

Durant la carrera he realitzat moltes assignatures enfocades a la realització de software, les quals m'han ajudat significativament per a realitzar aquest projecte.

Els primers sis capítols els he pogut desenvolupar gràcies a haver assistit a "Planificació i Gestió de Projectes i Sistemes Informàtics (PGPSI)". Allà es varen veure temes com l'abast d'un projecte, obstacles, metodologies de feina, eines de seguiment, Diagrames de Gantt, pla d'acció, estimació de costos, control de gestió i anàlisi de riscos.

A "Projecte d'enginyeria del software i bases de dades (PESBD)" vaig adquirir coneixements per a realitzar els casos d'us, esquema conceptual i disseny de models de base de dades.

A "Enginyeria del software I i II (ES1, ES2)" vaig aprendre com fer diagrames de classe de disseny i diagrames de seqüència.

Per la part de programació, les assignatures que més m'han ajudat han estat "Programació 1 (P1)" on em varen ensenyar a programar, "Projecte de programació (PROP)" on vaig aprendre a afrontar un problema i plasmar-ho a un programa organitzat amb classes. A "Bases de dades (BD)" vaig aprendre els conceptes bàsics de les bases de dades que en aquest projecte m'ha ajudat per al rendiment de l'aplicació a baix nivell.

Amb "Disseny de sistemes basats en la web (DSBW)" vaig aprendre a treballar en equip utilitzant control de versions del codi i com crear una API, utilitzar bases de dades no relacionals i a fer aplicacions *single page*. Per altra banda, a "Projecte de xarxes de computadors" ens varen ensenyar a treballar amb un projecte més complet, amb servidors, API, web i aplicació mòbil.

Finalment, de cara a la presentació podré demostrar el que vaig aprendre a "Habilitats directives i de comunicació (HDC)", on bàsicament apreníem a presentar en públic.

Tot i així, a la carrera he trobat a faltar algunes coses que m'ha aportat el món laboral. Tret de la experiència, tasques que he après, podrien ser com fer un codi organitzat i documentat, realitzar tests unitaris i de integració i el més important, com afrontar un projecte que necessita un manteniment constant ja que està en producció i molta gent depèn d'ell.

10.5 Treball futur

En aquest punt m'agradaria explicar quins plans de futur tinc amb aquest projecte.

El primer pas seria tenir una versió beta i posar-la en funcionament, tenir el primer contacte amb usuaris reals i recollir feedback. A partir d'aquest punt, l'escalabilitat pot ser necessària i

hauré d'estar preparat per a afrontar-la, com a mínim verticalment. També estic disposat a fer una inversió econòmica si es necessari.

Un cop obtingut el feedback i passat un temps des de la posada en marxa, apareixeran funcionalitats a millorar o a afegir. El més lògic seria començar amb les que tinguin més impacte sobre el major nombre d'usuaris. És evident que també apareixeran errors i possiblement es comencin a fer evidents els primers colls d'ampolla.

El següent pas seria ampliar la API Rest perquè tingui totes les funcionalitats de l'aplicació. El motiu d'aquest pas és crear les aplicacions natives mòbil, tant per Android com iOS. Potser arribats en aquest punt m'hauria de plantejar buscar a algú que es vulgui afegir al projecte i donar-me un cop de mà, ja que crear i mantenir tres plataformes és una tasca molt laboriosa per un únic desenvolupador.

El següent pas seria comercialitzar l'aplicació. Inicialment la idea era oferir-la als hotels de manera gratuïta i seguir així fins a obtenir certa reputació. No obstant, la part d'anuncis de comerços potser és pot monetitzar si ja disposem d'una ampla cartera d'usuaris.

Un punt important és fer-la multi idioma. L'aplicació ja conta amb els medis per ser-ho però faltaria buscar recursos per portar-ho a terme.

En paral·lel s'anirà millorant l'aplicació web perquè es pugui fer servir des de ordinadors de sobretaula o telèfons que no siguin Android o iOS.

Un cop el nombre d'usuaris sigui notable i l'aplicació comenci a generar ingressos (no beneficis), es pot començar a buscar inversors per tal d'ampliar l'equip i pensar en formar una petita empresa.

Finalment, seria bo que l'aplicació es convertís en un referent del sector i s'utilitzés mundialment.

11 Bibliografia

- [1] Amazon Web Services. [Online]. Disponible a <https://aws.amazon.com>. [Últim accés 10 agost 2016]
- [2] Ansible. [Online]. Disponible a <https://www.ansible.com/>. [Últim accés 10 agost 2016]
- [3] Bootstrap. [Online]. Disponible a <http://getbootstrap.com/>. [Últim accés 10 agost 2016]
- [4] Celery. [Online]. Disponible a <http://www.celeryproject.org/>. [Últim accés 10 agost 2016]
- [5] Classy CBV. [Online]. Disponible a <http://ccbv.co.uk/>. [Últim accés 15 agost 2016]
- [6] Classy DRF. [Online]. Disponible a <http://www.cdrf.co/>. [Últim accés 10 febrer 2016]
- [7] Daniel & Audrey Roy Greenfeld, «Two scoops of Django. Best practices for Django 1.8», Tercera edició, 2015.
- [8] Django extensions. [Online]. Disponible a <http://django-extensions.readthedocs.io>. [Últim accés 10 abril 2016]
- [9] Django Project. [Online]. Disponible a <https://www.djangoproject.com/>. [Últim accés 20 agost 2016]
- [10] Django Rest Framework. [Online]. Disponible a <http://www.django-rest-framework.org/> [Últim accés 5 febrer 2016]
- [11] Fever. [Online]. Disponible a <https://www.feverup.com/>. [Últim accés 20 juny 2015]
- [12] HelloTel. [Online]. Disponible a <http://www.hellotelapp.com/>. [Últim accés 20 juny 2015]
- [13] jQuery. [Online]. Disponible a <https://jquery.com/>. [Últim accés 10 agost 2016]
- [14] Metodología de desarrollo de software. [Online]. Disponible a https://es.wikipedia.org/wiki/Metodología_de_desarrollo_de_software. [Últim accés 5 setembre 2015]
- [15] NginX. [Online]. Disponible a <https://www.nginx.com/>. [Últim accés 10 agost 2016]
- [16] Peter Baumgartner i Yann Malet, «High performance django», *Lincon Loop*, Primera edició, 2014.
- [17] RabbitMQ. [Online]. Disponible a <https://www.rabbitmq.com/>. [Últim accés 10 agost 2016]
- [18] Stack Overflow. [Online]. Disponible a <http://stackoverflow.com/>. [Últim accés 10 agost 2016]
- [19] Tinder. [Online]. Disponible a <https://www.gotinder.com/>. [Últim accés 20 juny 2015]
- [20] TripAdvisor. [Online]. Disponible a <https://www.tripadvisor.es/>. [Últim accés 20 juny 2015]
- [21] uWsgi. [Online]. Disponible a <https://uwsgi-docs.readthedocs.io/en/latest/>. [Últim accés 10 agost 2016]
- [22] Varnish. [Online]. Disponible a <https://www.varnish-cache.org/>. [Últim accés 10 agost 2016]
- [23] Virtualenv. [Online]. Disponible a <https://virtualenv.pypa.io/en/latest/>. [Últim accés 12 gener 2016]

